

UNIVERSIDAD CARLOS III DE MADRID  
ESCUELA POLITÉCNICA SUPERIOR



GRADO EN INGENIERÍA MECÁNICA  
PROYECTO FINAL DE GRADO

**Aplicación Interactiva  
para la resolución de problemas de  
Mecánica de Máquinas**

Autor: Javier Miranda Puente  
Tutor: Cristina Castejón Sisamón

Leganés, junio de 2013

Título: Aplicación interactiva para la resolución de problemas de Mecánica de Máquinas.

Autor: Javier Miranda Puente

Director: Cristina Castejón Sisamón

## EL TRIBUNAL

Presidente: \_\_\_\_\_

Vocal: \_\_\_\_\_

Secretario: \_\_\_\_\_

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día \_\_ de \_\_\_\_\_ de 20\_\_ en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO

PRESIDENTE



# Agradecimientos

Quiero dar las gracias especialmente a las dos personas más importantes de mi vida y a las cuales les debo todo.

A mi padre por convertirme en la persona que soy ahora, con criterio y con una educación impagable. Estoy orgulloso de haber cumplido uno de sus deseos y que aún siga aquí, después de la tremenda enfermedad que sufrió, para dedicárselo. Espero poder seguir dedicándote muchos más de mis logros.

A mi madre, el pilar sobre el que se sustenta mi vida, incansable, siempre con una sonrisa, ha dedicado su vida a hacer la nuestra más fácil, sin pedir nada a cambio. Quiero agradecer enormemente todo lo que hace por mi cada día y dedicarle este proyecto por ayudarme y quererme incondicionalmente.

No quiero olvidarme de mis tres hermanos. Agradecer al que siempre me aguanta en casa con una paciencia infinita y me hace reír con su forma de ser única, sigue siendo así y espero que la vida te trate lo bien que te mereces. Gracias también a mi hermano mayor y a su mujer, a la que considero como una hermana, por ayudarme por animarme y por quererme tanto. Espero que todos vuestros planes futuros salgan bien y que pronto aumentéis esta familia de la que estoy tan orgulloso de pertenecer.

Muchas gracias.

# Resumen

Este proyecto de fin de grado se presenta como una solución a la dificultad que presentan los ejercicios de la asignatura de Mecánica de Máquinas. La metodología de resolución de cierto tipo de ejercicios resulta en muchas ocasiones poco intuitiva y difícil de asimilar.

Aprovechando el auge de las nuevas tecnologías y supliendo las necesidades del nuevo plan de estudios: Bolonia, se propone la creación de una aplicación digital interactiva donde se resuelvan de forma visual y fácilmente entendible, los ejercicios de la asignatura. Se empleará la herramienta de autor *Macromedia Flash*, especialmente diseñada para la creación de contenido web, y su lenguaje de programación *ActionScript*, con el que se dotará a la aplicación de un sistema de navegación a través de los diferentes pasos de resolución del problema. Todo ello teniendo en cuenta la normativa existente para el diseño software de aplicaciones e interfaces de usuario.

Esto se traducirá en un apoyo al profesorado, y una mayor autonomía del alumno en las horas dedicadas al trabajo personal de la asignatura. Se podrá acceder a la aplicación desde cualquier lugar con una conexión a internet.

Para aumentar la vida útil de la aplicación se ha establecido una metodología de creación de ejercicios, basada en plantillas y una guía de estilos, de tal forma que se pueda incrementar el número de ejercicios en la aplicación.

# Abstract

This degree Project is presented as a solution for the difficulty of the Machine Mechanics' exercises. The methodology to solve certain types of exercises is often unintuitive and difficult to assimilate.

With the rise of the new technologies and the needs of the new University plans: Bologna, it is proposed the creation of a digital interactive application which solved visually and easily understandable this type of exercises. It will be used the *Macromedia Flash* tool, which is specially designed for web content creation, and its programming language, *ActionScript*, with which the navigation system through the application will be created. It will be taken into account the existing standards for software and user interface design.

This will result into a teacher support, and greater learner autonomy for the learning process. The application is also accessible from anywhere with an internet connection.

To increase the application life it has been established a methodology for creating exercises, based on templates and a style guide, so that it will be possible to increase the number of exercises in the application.

# ÍNDICE DE CONTENIDO

---

<b>CAPÍTULO 1: Motivación y objetivos .....</b>	<b>1</b>
1.1) Introducción.....	2
1.2) Objetivos.....	3
1.3) Fases de desarrollo .....	4
1.4) Estructura del documento .....	5
 <b>CAPÍTULO 2: Estado del arte .....</b>	<b>7</b>
<b>I) La Enseñanza Universitaria Actual .....</b>	<b>8</b>
1.1) Proceso de Bolonia .....	8
1.2) Estructura de la Enseñanza Universitaria.....	10
1.3) Planes de estudio del primer ciclo, Grado .....	10
1.4) La nueva universidad .....	12
<b>II) Nuevas Tecnologías para la Enseñanza y el Aprendizaje .....</b>	<b>13</b>
2.1) Evolución de las TIC .....	13
2.2) Enseñanza Asistida por Ordenador (EAO) .....	14
2.2.1) Programas EAO según su estructura .....	14
2.2.2) Programas EAO según su finalidad.....	19
2.3) Enseñanza Virtual (e-learning) .....	29
2.4) La Web 2.0 y 3.0.....	31
2.4.1) Los portafolios digitales (E-Portfolios) .....	33
2.4.2) Las redes sociales .....	33
2.4.3) Documentos y aplicaciones en la red ( <i>Cloud Computing</i> ) .....	34
2.4.4) Objetos Digitales de Enseñanza-Aprendizaje (ODEA) .....	35
 <b>CAPÍTULO 3: Diseño de una aplicación interactiva .....</b>	<b>38</b>
<b>I) La Aplicación y su Interfaz de Usuario .....</b>	<b>39</b>
1.1) Ciclo de vida de una Aplicación Interactiva .....	39
1.2) Interfaz de Usuario .....	42

1.3) Evolución de las Interfaces de Usuario. ....	43
1.4) Estrategia de diseño de una IU .....	45
1.4.1) Mecanismos de interacción .....	46
1.4.2) Gestión de las entradas del usuario .....	50
1.4.3) Diseño gráfico de la aplicación .....	51
1.5) Prototipos .....	58
1.6) Evaluación.....	61
1.5) Conclusiones.....	63
<b>II) El usuario y el Entorno de la Aplicación .....</b>	<b>64</b>
2.1) Percepción visual .....	64
2.2) Cognición humana .....	65
Almacenamiento de la información. ....	66
Toma de decisiones.....	66
Prevención de los errores .....	67
2.3) Modelos mentales .....	68
2.4) El entorno del usuario.....	69
2.5) DCU (Diseño Centrado en el Usuario) .....	70
2.6) Conclusión .....	73
<b>III) Normativa y Estándares.....</b>	<b>74</b>
3.1) Introducción.....	74
3.2) Estándares .....	74
3.2.1) Estándares de iure para la IPO (Interacción Persona-Ordenador) .....	76
3.3) Guías de estilo.....	81
3.3.1) Guías de estilo comerciales.....	81
3.3.2) Guías de estilo corporativas .....	83
3.4 Conclusiones .....	84
 <b>CAPÍTULO 4: La Herramienta de Mecánica de Máquinas .....</b>	 <b>85</b>
<b>I) Análisis de requisitos .....</b>	<b>86</b>
1.1) La Asignatura de Mecánica de Máquinas.....	86
1.2) Dificultades de la asignatura .....	88
1.3) El alumno y su Entorno .....	94



<b>II) Diseño.....</b>	<b>95</b>
2.1) Descripción del funcionamiento .....	95
2.2) Entorno de la Herramienta y Normativa .....	96
2.3) Diseño e Implementación .....	99
2.3.1) Diseño de la IU .....	99
2.3.2) Diseño de la navegación .....	105
<b>III) Prototipos.....</b>	<b>111</b>
<b>IV) Evaluación .....</b>	<b>117</b>
<b>V) Planificación .....</b>	<b>121</b>
<b>VI) Lanzamiento.....</b>	<b>124</b>
 <b>CAPÍTULO 5: Conclusión y Futuros Desarrollos .....</b>	 <b>128</b>
<b>I) Futuros Desarrollos.....</b>	<b>129</b>
<b>II) Conclusiones.....</b>	<b>131</b>
 Bibliografía.....	 132
 <b>ANEXO I: Manual de Usuario.....</b>	 <b>135</b>
Enunciado .....	136
Resolución .....	137
Final del Paso .....	138
Final del Ejercicio .....	139
 <b>ANEXO II: Metodología de creación de ejercicios .....</b>	 <b>140</b>
Interfaz de Usuario .....	141
Navegación .....	145
 <b>ANEXO III: Formulario de Evaluación.....</b>	 <b>148</b>

# ÍNDICE DE FIGURAS

---

Figura 1: Países integrantes en el proceso de Bolonia y su año de incorporación [6]	9
Figura 2: Estructura de un programa lineal	15
Figura 3: Módulos que incorpora un Sistema Tutorial Inteligente [9]	17
Figura 4: Ejemplo de Herramientas de autor. Silverlight, Tumult Hype, Squeak y Macromedia Flash (de izquierda a derecha)	22
Figura 5: Programas de autor dedicados a la enseñanza. Hot Potatoes, JClic, Exe Learning, EdiLim, Lams, Mated (de izquierda a derecha) [15]	23
Figura 6: Interfaz de usuario de Adobe Flash Professional CS5	25
Figura 7: Ejemplo del resultado de Capa máscara + Capa enmascarada [14]	26
Figura 8: Ventana de "Acciones". Panel de código (Derecha), Panel de Acciones (Izquierda superior), Listado de fotogramas con código (Izquierda inferior).	28
Figura 9: Ciclo de vida de un Sistema Interactivo	41
Figura 10: Elementos de un Sistema Interactivo	42
Figura 11: Consola de Windows basada en Interfaz de Línea de Mandatos (CUI)	43
Figura 12: Interfaz de Menú de Windows	44
Figura 13: Evolución en la interfaz de Windows. Escritorio de Microsoft Windows XP (atrás), Microsoft Windows 7 (delante.)	44
Figura 14: Interfaz Web. Metáfora de una tienda	45
Figura 15: Estructura de una IU: "look-and-feel iceberg" de IBM (1992)	45
Figura 16: Mecanismos gráficos de introducción de valores	47
Figura 17: Menú de selección. (Mecanismo de selección de tamaño fijo y elementos invariables)	48
Figura 18: Ventana contenedora de archivos (Ejemplo de mecanismo de selección de tamaño variable)	48
Figura 19: Barra de herramientas con botones icónicos	49
Figura 20: Lista desplegable, selección de fuente Microsoft Word	49
Figura 21: Selección de opciones de texto	49
Figura 22: Mecanismo de arrastre de una ventana de Windows	50
Figura 23: Representación de las partes de una aplicación interactiva	51
Figura 24: Principios del diseño visual [37].	52
Figura 25: Reglas de Claridad Visual de Gestalt.	53
Figura 26: Iconos de navegación grandes	53
Figura 27: Botones de ventana en el borde de la misma	53
Figura 28: Ejemplo de falta de intuitividad.	54
Figura 29: Correspondencia entre iconos y acciones de los botones	55
Figura 30: Interrupción del flujo de trabajo debido a una ventana de información	56
Figura 31: Diferencia entre una mala estructuración y uso de colores (Izquierda), y una IU con una estructura legible y clara (Derecha).	56
Figura 32: Ejemplo de prototipo StoryBoard para una página Web	60
Figura 33: Ejemplo de prototipos software	60
Figura 34: La atención se ve guiada al círculo rojo porque es el único inusual. Fuente: <a href="http://www.human-computer.net/blog">www.human-computer.net/blog</a>	65
Figura 35: Mayor limitación en la opción de abajo.	67
Figura 36: Sugerencias en la búsqueda.	67
Figura 37: Mensaje de confirmación de GMai.	67
Figura 38: Corrección en la búsqueda de Google	67

Figura 39: Distintos modelos mentales sobre un mismo objeto	68
Figura 41: Ejemplo de convenciones para la búsqueda (las dos primeras)	69
Figura 40: Extracción de un mapa conceptual a partir de un texto	69
Figura 42: Planos y elementos en el diseño centrado en el usuario (según Garret, 2002) [40].	71
Figura 43: Esquema de la labor del diseñador y del usuario.	71
Figura 44: Mapas de calor sobre tres interfaces diferentes. Fuente: Nielsen (año 2006) [37].	72
Figura 45: Logotipos de los comités y organizaciones más importantes de creación de estándares y normas	75
Figura 46: Guía de estilo de Apple Corporation	81
Figura 47: Guía de estilo IBM (1992)	82
Figura 48: Guía de estilo Microsoft (2000)	82
Figura 49: Guía de estilo Motif (Izquierda) y Java (Derecha)	82
Figura 50: Proceso para la creación de una Guía de estilo de un producto. Se comenzará desde abajo y se irá ascendiendo por la pirámide cumpliendo cada fase.	83
Figura 51: Representación gráfica de los CIR de un mecanismo [44]	89
Figura 52: Obtención de los vectores velocidad de una estructura a partir del Cinema de Velocidades [44]	90
Figura 53: Determinación gráfica de la aceleración de un punto [44]	91
Figura 54: Ejemplo del método gráfico de reducción de fuerzas [44].	93
Figura 55: Página web " <a href="http://maqlab.uc3m.es/TMM/tmm.html">http://maqlab.uc3m.es/TMM/tmm.html</a> " [45] desde el navegador Mozilla Firefox	97
Figura 56: Representación de los botones dispuestos sobre una "mesa", ordenados y con presentación icónica	100
Figura 57: Interfaz de usuario de la aplicación de Mecánica de Máquinas (metáfora "Pizarras")	101
Figura 58: Nueva apariencia de los elementos de una estructura	102
Figura 59: Tipos de símbolos en Macromedia Flash.	106
Figura 60: Línea de tiempo de un Botón.	106
Figura 61: Estructura de la línea del tiempo de la capa "as"	107
Figura 62: Prototipo número 1 de la IU.	112
Figura 63: Prototipo número 2 de la IU.	112
Figura 64: Prototipo número 3 de la IU.	113
Figura 65: Prototipo número 4 de la IU	114
Figura 66: Maqueta digital estática que representa la plantilla final.	115
Figura 67: Prototipo número 1 de la navegación	116
Figura 68: Evaluación de los aspectos visuales de la aplicación.	118
Figura 69: Evaluación de los aspectos interactivos de la aplicación.	118
Figura 70: Evaluación de los aspectos pedagógicos de la aplicación (Muestra completa, 11 encuestados)	118
Figura 71: Evaluación de los aspectos pedagógicos de la aplicación (Muestra parcial, 7 encuestados con conocimientos previos de la asignatura).	119
Figura 72: Diagrama circular de satisfacción general con los aspectos visuales de la aplicación	120
Figura 73: Diagrama circular de satisfacción general con los aspectos interactivos de la aplicación	120
Figura 74: Diagrama circular de satisfacción general con los aspectos pedagógicos de la aplicación	120
Figura 75: Planificación del proyecto mediante diagrama de Gantt	123
Figura 76: Previsiones sobre las tecnologías en auge para los próximos cinco años. Informe Horizon 2012 (Izquierda) [2]; Informe Horizon 2013 (derecha) [3].	129
Figura 77: Organización de las capas y la línea del tiempo de la plantilla para los problemas.	141
Figura 78: Línea del tiempo del botón "+detalles".	144

# ÍNDICE DE TABLAS

---

<i>Tabla 1: Comparación entre los diferentes tipos de programas según su estructura</i>	18
<i>Tabla 2: Comparación entre las herramientas de autor y los programas de autor dedicados a la enseñanza</i>	24
<i>Tabla 3: Características de la formación presencial y en red (Cabrero et al., 2005) [20]</i>	29
<i>Tabla 4: Ventajas de los objetos de aprendizaje. Fuente [31]</i>	36
<i>Tabla 5: Estados implicados en el proceso de desarrollo de un estándar ISO</i>	76
<i>Tabla 6: Elementos de la navegación.</i>	108
<i>Tabla 7: Otros elementos que intervienen en la programación de la navegación.</i>	109
<i>Tabla 8: Planificación por fechas del análisis, diseño y creación de la Aplicación de Mecánica de Máquinas</i>	121
<i>Tabla 9: Presupuesto para la aplicación de Mecánica de Máquinas</i>	127

# ***CAPÍTULO 1***

## **MOTIVACIÓN Y OBJETIVOS**

---

## 1.1) Introducción

La tecnología ha evolucionado durante las últimas décadas de forma exponencial y con una repercusión muy fuerte en la sociedad. Los sistemas tecnológicos han incurrido en todos los ámbitos de la vida cotidiana del ser humano: la comunicación, el trabajo, las finanzas, el mercado, la educación y aprendizaje, etc. El motivo que ha impulsado a las tecnologías a su continuo desarrollo es la manera en que facilitan todos estos procesos.

El presente proyecto pretende transportar las ventajas que ofrecen los sistemas tecnológicos existentes, al ámbito de la educación en las Universidades. Concretamente al estudio práctico de la asignatura de Mecánica de Máquinas, presente en el programa del Grado en Ingeniería Mecánica impartido por la Universidad Carlos III de Madrid.

Aprovechando la inclusión de esta Universidad en el nuevo modelo de estudios europeos, que propone un aprendizaje más autónomo por parte del alumno, se crea una aplicación interactiva de resolución de problemas que le ayude en esta tarea autodidacta.

La motivación para realizar este proyecto surge de la experiencia y el conocimiento tanto de la tutora como del desarrollador de la aplicación, en la materia de la asignatura. Se tratan problemas con un gran contenido gráfico (estructuras en 2D, vectores, cinemas de velocidades y aceleraciones, etc.) y se usa una metodología de resolución sencilla pero muy visual. El principal problema es que hasta el momento estos ejercicios solo están resueltos sobre papel y con figuras estáticas. En el año 2005 se presentó un proyecto titulado “Aplicación de nuevas tecnologías orientadas al aprendizaje de Teoría de Máquinas y Mecanismos” [1], que proponía una página web que reuniese la teoría y los problemas de dicha asignatura.

Con la evolución de las herramientas de autor para la creación de animaciones y sistemas interactivos, este proyecto propone el diseño de una ODEA (Objeto Digital de Enseñanza y Aprendizaje) con una estructura ramificada y sencilla que permita al alumno navegar entre los diferentes pasos del problema, estableciendo así su propia velocidad de aprendizaje. A su vez, se ha dotado a la aplicación de animaciones dinámicas e intuitivas mejorando de este modo la herramienta predecesora. Todo ello se ha conseguido mediante la potente herramienta *Adobe Flash Professional*, especializada en la creación de sistemas interactivos web.

La finalidad de este proyecto se puede resumir en la intención de facilitar el proceso de aprendizaje autónomo del alumno y complementar la labor del profesorado, apostando por una aplicación interactiva sencilla e intuitiva, con acceso online, que ayude a asimilar la resolución de los problemas de la asignatura de Mecánica de Máquinas.

## 1.2) Objetivos

La idea fundamental que guiará este proyecto es facilitar la comprensión de los problemas de la asignatura de Mecánica de Máquinas que se resuelven mediante métodos gráficos. Se intentará suplir las carencias de la metodología empleada hasta ahora, basada en la enseñanza mediante dibujos estáticos en papel o diapositivas, y mejorar el contenido web desarrollado en 2005 por un proyecto sobre esta asignatura.

Para ello se proponen diversos objetivos parciales que serán estudiados a lo largo de este documento y que tienen como trasfondo el objetivo principal previamente propuesto:

1) Creación de una aplicación interactiva que resuelva paso a paso aquellos ejercicios de la asignatura que presenten una metodología gráfica. Para ello se ha de conseguir una animación clara y visual de los elementos que pueda ser comprendida por cualquier alumno. Además será necesario establecer un método de resolución de los ejercicios que sea lo más sencillo y entendible posible.

2) Diseñar una interfaz de usuario organizada y amena que sea fácilmente reconocible por el alumno y que cumpla con la normativa específica para este campo.

3) Programar un sistema de navegación a través de los pasos del problema, intuitivo y sencillo pero a la vez robusto, que permita establecer la velocidad de aprendizaje deseada por el alumno y que no genere errores inesperados.

4) Permitir el acceso al contenido desde cualquier lugar con conexión a internet. La aplicación se alojará en la página web existente creada para el antiguo proyecto. Los archivos asociados serán compatibles con cualquier navegador incluso desde dispositivos móviles, siempre que permitan *Adobe Flash Player*.

5) Establecer un patrón y una metodología para la creación de ejercicios de la asignatura. Se proporcionarán las plantillas, las guías de estilo, el código para la navegación y los métodos de resolución de cada tipo de problemas para aumentar el repertorio en futuros desarrollos.

## 1.3) Fases de desarrollo

El nacimiento del proyecto comenzó con la necesidad de cumplir el objetivo principal anteriormente expuesto. La idea de facilitar el proceso de aprendizaje de cierto tipo de problemas desencadenó el estudio realizado en este documento y terminó con la aplicación de Mecánica de Máquinas. Las fases que han tenido lugar durante este proceso se exponen a continuación:

**Fase 1:** La idea requirió el estudio del estado de la enseñanza y la tecnología actual. Fue importante conocer las pautas del nuevo sistema educativo para mantener la calidad de enseñanza y adaptar la aplicación a la nueva Universidad. Entre los diversos cambios que propone el EEES (Espacio Europeo de Educación Superior), la importancia dada al trabajo personal del alumno, será uno de los pilares sobre los que se fundamenta este proyecto.

Así mismo, fue necesario hacer un estudio de los sistemas tecnológicos dedicados a la enseñanza. Se muestran las diferentes opciones para llegar al objetivo propuesto: aplicaciones con diferentes estructuras, diversas herramientas para su creación y los métodos de divulgación por la web. Se hicieron comparativas y se justificó la elección que mejor se ajustaba al objetivo inicial.

**Fase 2:** Después de concluir que la opción más viable era la creación de una aplicación interactiva, hubo que buscar toda la información referente a su diseño, al entorno donde se crearía y a la normativa referente a su creación. Para desarrollar una aplicación dedicada a la enseñanza es de vital importancia seguir una estrategia de diseño, saber quién la utilizará, para adecuar su diseño al usuario y seguir las directrices establecidas por la normativa y las guías de estilo existentes.

**Fase 3:** Se llevó a cabo un análisis de los requisitos de la aplicación antes de su creación. Fue necesario estudiar el contenido de la asignatura para establecer el método de resolución de los problemas. Se analizaron las dificultades que presentaba la asignatura y se hizo un estudio del perfil del usuario de la aplicación.

**Fase 4:** Con toda la información recopilada y con las elecciones tomadas, se comenzó a diseñar la aplicación. Hubo que aprender a usar la herramienta de autor (*Adobe Flash CS5*) y su lenguaje de programación (*ActionScript*) mediante manuales y tutoriales. Se determinó el funcionamiento, el estilo y la presentación que se pretendía de la aplicación. Se crearon prototipos para estudiar las ventajas, los inconvenientes, las posibles mejoras y la aceptación entre algunos usuarios.

**Fase 5:** Se llevó a cabo la implementación de la interfaz de usuario y de la navegación en la aplicación. Se crearon los diferentes problemas, teniendo en cuenta todas las pautas establecidas hasta el momento, y se alojaron en el dominio web de la antigua aplicación.



**Fase 6:** Con la aplicación ya desarrollada se elaboró un presupuesto en base a la planificación seguida. Se realizó un proceso de evaluación para determinar los puntos fuertes y débiles de la aplicación y para asegurar el cumplimiento de los objetivos. Además se establecieron las plantillas y la guía de estilo del producto para futuros desarrollos. Por otro lado, fue necesaria la creación de un manual de usuario para explicar la utilización y todos los detalles de la aplicación.

**Fase 7:** Por último se enumeró una lista de posibles vías de desarrollo que se pueden llevar a cabo en un futuro, para ampliar las capacidades y las ventajas que ofrece esta aplicación.

### 1.4) Estructura del documento

Para facilitar la lectura de la memoria se hará un breve resumen del contenido de cada capítulo:

**Capítulo 1:** Describe el contexto general del proyecto, explicando cómo surge la idea en la que se basa y los objetivos que se desean conseguir. Se detalla el proceso seguido desde el inicio hasta la obtención del producto final y se presenta la estructura del documento.

**Capítulo 2:** Recopila la información necesaria sobre el marco educativo donde se establecerá el producto, así como la información sobre la tecnología dedicada a la enseñanza existente hasta el momento. También se presenta la evolución de la enseñanza virtual y su relación con el auge de la Web 2.0 y 3.0.

**Capítulo 3:** Se divide en tres partes: la primera establece las pautas necesarias para crear una aplicación interactiva, su interfaz gráfica y los métodos de interacción; la segunda determina el análisis del usuario, su percepción de la aplicación y establece los principios de un diseño centrado en el alumno; la tercera parte muestra las directrices, la normativa y las guías de estilo aplicables a las aplicaciones interactivas.

**Capítulo 4:** Se presenta el proceso seguido para la creación de la aplicación. Los análisis previos, el diseño, la implementación del producto final, y su lanzamiento. Se presentan todos los prototipos generados a lo largo del proceso, así como la planificación y el presupuesto final.

**Capítulo 5:** Se establecen las líneas de futuros desarrollos según los informes *Horizon* 2012 [2] y 2013 [3] sobre la enseñanza universitaria y las tecnologías en auge.

**Bibliografía:** Muestra la bibliografía utilizada en el presente documento.

**Anexo I: Manual de usuario:** Proporciona una ayuda al usuario que utilice por primera vez la aplicación. En este manual se pueden encontrar todos los elementos de interacción y navegación que ofrece la aplicación, así como la organización de la interfaz y la estructura de los pasos.

**Anexo II: Metodología de creación de ejercicios:** Dirigido a aquellos profesores o usuarios que deseen aumentar el repertorio de ejercicios. Se proporciona una guía de estilos de la aplicación y la metodología a seguir para desarrollar nuevos ejercicios para la aplicación.

**Anexo III: Formulario de Evaluación:** Se presenta el formulario utilizado para el estudio de calidad de la aplicación previo al lanzamiento de la misma. Este formulario se utiliza para detectar los puntos fuertes y débiles de la interfaz, la interacción y los aspectos pedagógicos de la aplicación.

# ***CAPÍTULO 2***

## **ESTADO DEL ARTE**

---

# I) LA ENSEÑANZA UNIVERSITARIA ACTUAL

## 1.1) Proceso de Bolonia

El proceso de creación de los estudios oficiales de Grado nace ante la necesidad de armonizar los sistemas de educación superior en Europa. Así, en 1998 con la iniciativa de Francia, Reino Unido, Alemania e Italia, se realiza la Declaración de la Sorbona. Con esto se pone en marcha la creación de un Espacio Europeo de Educación Superior (EEES) que favorece el desarrollo de la Europa del conocimiento. [4]

La consolidación de este proceso se da en 1999 con la Declaración de Bolonia firmada por 29 países, entre ellos España. Se fijaron los primeros objetivos para cumplir en 2010:

- Instauración de una serie de titulaciones flexibles, comprensibles y comparables, mediante la implantación de un Suplemento Europeo al Título (SET) que promuevan un mejor acceso al trabajo y una mayor competitividad internacional para el estudiante.
- Un sistema de estudios estructurado en tres ciclos: Grado, Máster y Doctorado.
- Una unidad de medida del trabajo realizado por el estudiante: el crédito ECTS (*European Credit Transfer System*).
- Una mayor facilidad de movilidad europea e internacional por parte del estudiante.
- Un nivel alto de calidad tanto de las nuevas titulaciones como de las instituciones universitarias que las imparten.
- Un balance entre este nivel de calidad y el económico para facilitar el acceso a cualquier estudiante independientemente de sus posibilidades.

En el territorio nacional, el Real Decreto 49/2004 del 19 de Enero [5], establece los requisitos para definir los nuevos títulos universitarios específicos de Grado. De esta forma las universidades podrán elaborar sus planes de estudios propios dentro de las directrices establecidas. Podrán dar más o menos importancia a aspectos como la orientación profesional o la preparación del estudiante en temas más prácticos, entre otros. Esto se consigue con itinerarios de libre configuración curricular para el alumno.

En la siguiente Figura 1 se muestran los 46 países participantes en el proceso de Bolonia y su año de incorporación.



## 1.2) Estructura de la Enseñanza Universitaria

Las condiciones de la estructura de estas enseñanzas vienen dispuestas en la Ley Orgánica 6/2001 de 21 de diciembre, de universidades y en el Real Decreto del 29 de mayo del 2003 [5].

Los estudios de educación superior se organizan en los ciclos de Grado, Máster y Doctorado.

El primer ciclo, correspondiente al nivel de Grado, comprende una formación básica y general orientada a la actividad profesional. El título se obtendrá siguiendo un plan de estudios determinado por cada universidad bajo los requisitos que establece el Real Decreto. Así los diferentes planes para los Grados han de superar rigurosos controles de evaluación interna y externa para asegurar la calidad exigida. Uno de los requerimientos es la realización de 240 créditos ECTS que se distribuyen en cuatro años para los títulos oficiales de Licenciado, Ingeniero o Arquitecto.

El segundo ciclo, correspondiente a los estudios de Máster, está dedicado a la especialización de los conocimientos adquiridos en el primer ciclo o a la preparación para la investigación en un determinado campo. Comprenderán un total de entre 60 y 120 créditos ECTS que se impartirán en uno o dos años.

El tercer ciclo, correspondiente al Doctorado, consiste en la realización de un periodo de investigación que finaliza con la defensa de una tesis doctoral.

El segundo y tercer ciclo reciben el nombre de Postgrado. El plan académico de estos dos niveles lo establece también la Universidad ajustándose a la normativa específica reguladora aprobada por el Gobierno. Los planes podrán cambiar de un año para otro siempre y cuando se trate de mejoras en la formación del estudiante.

## 1.3) Planes de estudio del primer ciclo, Grado

El plan de estudios es el conjunto de enseñanzas organizadas por una universidad. Dentro del mismo se distinguen varios tipos de materias a las cuales se asignan un número de créditos determinado.

El crédito es una unidad de medida de la actividad académica del alumno en la que se valoran: las enseñanzas teóricas y prácticas, las actividades académicas dirigidas y el trabajo personal del alumno. Un crédito comprenderá entre 25 y 30 horas. Existen varios tipos de materias que deben distinguirse en todo plan de estudios [6]:

- Materias troncales: han de seguir estrictamente las directrices generales propias del título y no podrán denominarse de forma distinta a lo establecido. Todas tendrán el mismo número de créditos.
- Materias definidas por la Universidad: pueden ampliar los conocimientos de las materias troncales o tener un objetivo formativo diferente. Estas materias se subdividen en:
  - Materias obligatorias que serán definidas libremente por la universidad y de obligada elección por los alumnos. Tendrán hasta un máximo del 30 por ciento del total de créditos del plan de estudios.
  - Materias optativas. Igualmente libre definidas por la universidad, sin embargo el alumno podrá elegir cuales cursará dependiendo de sus preferencias. Tendrán hasta un máximo del 20 por ciento del total de créditos del plan de estudios.
  - Créditos de libre elección. Se trata de un porcentaje de créditos que harán referencia a materias de otros planes, seminarios u otras actividades académicas ofertadas por la universidad y entre las cuales el alumno podrá elegir. Tendrán hasta un máximo del 5 por ciento del total de créditos del plan de estudios. Obligatoriamente, el contenido de las enseñanzas que escoja el alumno no puede tener relación con las materias propias de la titulación correspondiente. De esta forma se asegura que el alumno adquiera conocimiento en otras áreas completamente diferentes.

La forma de evaluar cada asignatura dependerá de su contenido pero seguirá ciertas directrices. Hay diversos factores que cuentan para la evaluación, no solo los exámenes sino también trabajos realizados fuera de clase o las prácticas externas en empresas. Así surgen dos modalidades de evaluación:

- Evaluación continua donde se realiza un seguimiento constante del alumno mediante trabajos y evaluaciones. Para ello se proponen dos herramientas: el potencial que ofrece internet y las nuevas tecnologías de información y comunicación (TIC); las tutorías personales.
- Enseñanza práctica donde se busca una participación activa del alumno en el aprendizaje.

Las horas dedicadas a la docencia tanto teórica como práctica no excederá del 50 por ciento del total de créditos de la materia. Por lo tanto es importante destacar que un porcentaje del total de horas mide el trabajo individual. Todo el tiempo invertido en el aprendizaje se valorará en créditos europeos. Este es uno de los principales cambios con respecto a los planes académicos antiguos por el cual se incita al alumno al auto-aprendizaje. De esta forma, las herramientas como la que estudia este proyecto, adquieren una elevada importancia aportando una ayuda muy beneficiosa en el proceso de aprendizaje individual.

## 1.4) La nueva universidad

Las universidades españolas ofrecen titulaciones clásicas como Medicina, Derecho e Historia, pero además son capaces de ofertar otras más novedosas como Ingeniería de la Edificación, Aeroespacial, etc. Las universidades son libres de configurar planes de estudio nuevos siempre que sigan unos requisitos de calidad establecidos. En España, por ejemplo, la encargada de aprobar finalmente dichos planes es la Agencia Nacional de Evaluación de la Calidad y Acreditación (ANECA).

Por otro lado, con el nuevo sistema educativo, el proceso de aprendizaje da más relevancia a la labor del alumno que a la del profesorado. Se valoran más las habilidades y capacidades generales y específicas, lo cual será decisivo para acceder al mercado laboral. También incentiva una forma de estudio autónoma, más reflexiva y multidisciplinar, basada en la cooperación y en la práctica. Una mayor autonomía del alumno se traducirá en el futuro como una mayor capacidad para resolver problemas y para tener un pensamiento crítico.

Otra de las ventajas de la nueva universidad es la fácil movilidad tanto nacional como internacional del estudiante, así como la flexibilidad en el reconocimiento de los créditos superados. Este es uno de los objetivos principales del EEES. Un estudiante que realice el primer ciclo en un país europeo, tendrá completa libertad para realizar el segundo y tercer ciclo en otro país cualquiera. Se reconoce el trabajo realizado de una forma equitativa en todos los países de la Unión Europea.

En definitiva el proceso de Bolonia consiste en una reforma pedagógica y de organización de los estudios universitarios. Esto se lleva a cabo mediante unos mecanismos y normas que aseguran un elevado nivel de calidad de enseñanza, bajo un marco europeo que lo controla.

En España, desde el año 2010, todas las universidades tienen ofertados los títulos de grado y en particular la Universidad Carlos III, pionera en establecer los nuevos planes, ya ha tenido los primeros egresados.



## II) NUEVAS TECNOLOGÍAS PARA LA ENSEÑANZA Y EL APRENDIZAJE

### 2.1) Evolución de las TIC

El auge de la tecnología comenzó en el 1880. Se produjo un cambio en el modo de enseñanza, que surgió con la aparición de materiales como las filminas, diapositivas y películas. Más tarde con la Primera Guerra Mundial se exigió una formación rápida y eficiente volviendo a provocar cambios sustanciales en los conceptos educativos [7].

La televisión aparece oficialmente entre los años 1923 y 1933. Posteriormente, en el año 1945 empezaron a registrarse las primeras aplicaciones de esta tecnología relacionadas con la educación.

Otro de los hitos significativos de la evolución de la tecnología se produjo el 4 de octubre de 1957 con el lanzamiento del primer satélite artificial llevado a cabo por la Unión Soviética. En EEUU se comienza a introducir gran cantidad de dispositivos técnicos en los sistemas educacionales y se modifican los planes de estudio. Surgen los laboratorios de idiomas audio-activos comparativos, la televisión en circuito cerrado con video-grabadoras domésticas, y la enseñanza asistida por ordenador.

Fue a principios de los 60 cuando los ordenadores empezaron a extenderse por las universidades siendo su uso cada vez más relevante con el paso del tiempo. Su desarrollo y evolución ha ido creciendo a pasos agigantados hasta el día de hoy. Cada vez se diseñan ordenadores más pequeños y con mayor capacidad de almacenamiento y procesamiento. Se ha llegado al punto de fabricar dispositivos tan pequeños que caben en un bolsillo y con una potencia de cálculo mucho superior a sus predecesores. Además se crean interfaces más intuitivas manteniendo las prestaciones, para llegar a un público más amplio. Aún así, la tecnología tiene mucho que evolucionar y lo seguirá haciendo hasta llegar a límites todavía impensables.

La principal característica de las NTIC es la desmaterialización, deslocalización y globalización de la información. Las universidades implantan continuamente y de forma creciente nuevas tecnologías que apoyan a la docencia. Un estudio realizado en 2008 en España [8], concluyó que la media de alumnos por ordenador en las aulas de docencia era de 14.6, el 81% de las aulas poseían cobertura “wifi” y un 52.1% de las asignaturas poseían una plataforma software de apoyo a la docencia. Hoy en día, la cantidad de aulas dedicadas a la

informática ha crecido notablemente y se hace raro encontrar una universidad sin posibilidad de conexión a internet y sin una plataforma web para compartir información.

Esta evolución se puede apreciar en todos los niveles de la educación, tanto a nivel de conceptos básico (Por ejemplo la sustitución de las pizarras por presentaciones proyectadas) como a niveles especializados con el uso de potentes herramientas de cálculo y análisis.

Sin embargo la tecnología que más influencia tiene en la enseñanza actualmente es el uso del ordenador, y ligado a ello, el uso de internet. En este apartado se explicarán los principios de la enseñanza asistida por ordenador (EAO) y la evolución de internet hasta su punto más alto en la actualidad con la web 2.0 y 3.0.

## 2.2) Enseñanza Asistida por Ordenador (EAO)

El concepto de EAO engloba a aquellos programas o sistemas informáticos contruidos y diseñados con finalidades educativas. Existe gran cantidad de ejemplos actualmente que van desde los “programas conductistas”, basados en secuencias lineales de contenidos y evaluación, hasta las sofisticadas aplicaciones interactivas actuales, dotadas de la denominada “navegación transversal”. Por otro lado, se pueden clasificar por su contenido o finalidad, ya sea: facilitar el trabajo del usuario, ayudarle a aprender mediante ejercicios de repetición, simular entornos de trabajo, etc.

Todos los tipos de programas pertenecientes a la EAO, se definen a continuación mediante dos clasificaciones: su estructura y su finalidad.

### 2.2.1) Programas EAO según su estructura

Los primeros sistemas de enseñanza en los que estaba implicado el uso de un ordenador, aparecieron en los años 50 y se denominaron “Programas lineales”. Se trataba de una programación muy básica que obligó a buscar nuevos métodos que solucionasen las deficiencias que presentaban. Así surgieron los “Programas ramificados”, y los “Sistemas generativos” [9].

Estos tres tipos de programas se denominaron CASI (*Computer Assisted Instruction*) y presentaban una deficiencia común: la independencia entre el contenido de la enseñanza y la forma de hacerlo.

De esta forma, se evolucionó hasta la creación de los sistemas ITS (*Intelligent Tutor Systems*) que se trata de programas informáticos (*software*) que combinan modelos psicológicos del estudiante y del profesor, teorías de educación y, el factor más importante, la IA (Inteligencia Artificial).

### Programas lineales

Este tipo de programas se caracterizan por tener un orden fijo en la enseñanza, establecido por el programador. El problema que ocasiona esta “linealidad” es que no se contempla la posibilidad de que el alumno no entienda los conceptos que expone el programa. Esta delimitación tiene su origen en la teoría conductiva de B.F. Skinner (1950) que aseguraba que el individuo funcionaba por estímulos, y dependiendo de cuáles fueran estos, se obtenía una respuesta u otra.

Por otro lado, los programas lineales imparten una enseñanza general, es decir, todos los individuos reciben el mismo conocimiento y en la misma secuencia. Esta enseñanza no individualizada, no tiene en cuenta que un individuo tenga más problemas para entender las explicaciones, que necesite más ejemplos, etc. Es decir, no se contemplan las particularidades de cada individuo. La ventaja que aporta es su sencilla programación.

La estructura de este tipo de programas está formada por tres partes (Figura 2): salida del programa, entrada del alumno y reacción del programa.



*Figura 2: Estructura de un programa lineal*

### Programas ramificados

Se trata de los sucesores de los programas lineales. También están compuestos por una secuencia fija y un número determinado de temas, sin embargo añaden la posibilidad de actuar dependiendo de la respuesta del individuo. Es la llamada capacidad de retroalimentación, que permite al individuo trabajar con el programa al ritmo que crea conveniente.

Esta capacidad de actuar ante las diferentes respuestas se consiguió gracias a la técnica “*Pattern-Matching*”. Dicha técnica permite clasificar una respuesta como aceptable o parcialmente aceptable, mejorando la técnica de B.K. Skinner que las clasificaba en totalmente correctas o incorrectas.

La ventaja de estos programas es la capacidad de ajustar el temario según las necesidades del individuo. Se consigue mediante la posibilidad de repetir la explicación, los textos, los ejercicios, etc. Sin embargo sigue proporcionando una enseñanza generalizada sin atender a las particularidades de cada usuario.

### **Sistemas generativos**

Este tipo de programas surge entorno a 1967-1971 con la filosofía de que un individuo aprende de una forma más eficaz enfrentándose a problemas que se adaptan a sus capacidades y no siguiendo un programa con una secuencia inmodificable [9]. En definitiva se pretende adaptar la enseñanza a las particularidades individuales, objetivo que no se observaba en programaciones anteriores.

El fundamento del que parte, es que el material de enseñanza, en este caso el ordenador, debe ser capaz de generar los problemas y sus soluciones dependiendo de las respuestas del individuo. Debe ser capaz de determinar el nivel de dificultad del problema a plantear, recibir la respuesta del alumno y compararla con la solución que ha generado.

Algunos de los inconvenientes que presenta esta filosofía de programación son: la existencia de materias que plantean problemas con un nivel de dificultad demasiado grande para ser generado por el ordenador y la existencia de varias soluciones válidas para un mismo problema.

### **Sistemas tutoriales inteligentes (ITS)**

Este tipo de programación se fundamenta en la filosofía de actuar como lo haría un profesor. El sistema debe enseñar unos conocimientos teniendo en cuenta el nivel del individuo durante todo el proceso de aprendizaje, e incluso debe ser flexible y estar abierto ante sugerencias preguntas y modificaciones.

La estructura de un ITS es compleja y acepta varios caminos para implementar un mismo programa. Sin embargo, todo programa presenta unos módulos fijos que pueden observarse en la Figura 3.



*Figura 3: Módulos que incorpora un Sistema Tutorial Inteligente [9]*

#### *Módulo dominio*

Es la base del programa, donde se encuentra la materia que será enseñada al alumno. Debe ser lo suficientemente extensa como para abordar todas las preguntas que pueda tener el individuo. Además debe poder ser actualizable si se quiere que el programa perdure frente a la evolución de la materia.

#### *Módulo estudiante*

Contiene la información de la materia específica de enseñanza. Esta información puede variar desde, por ejemplo, el cálculo de las respuestas incorrectas, hasta una complicada estructura que intente representar los conocimientos del alumno sobre el tema.

Cuanta más precisa sea la información sobre los conocimientos que posee el alumno, más personalizada será la enseñanza. Además precisará una actualización incremental a medida que vayan aumentando las capacidades del individuo.

#### *Módulo pedagógico*

Este módulo busca la forma más efectiva de realizar el proceso de aprendizaje. Depende mucho de la materia de enseñanza. Un programa será más efectivo, cuanto mejores resultados dé en la enseñanza del individuo. La mejor forma de llevar un conocimiento a un individuo nunca es única, y en la mayor parte de los casos depende de las capacidades del propio individuo.

### *Módulo interfaz*

Es el módulo que pone en contacto al alumno con el programa. Es lo que el alumno ve y con lo que puede interactuar, introduciendo su respuesta. Debe ser intuitivo y dinámico de tal forma que llame la atención del individuo.

## **Comparaciones entre los diferentes tipos de programas EAO según su estructura**

En la Tabla 1 se presenta un cuadro resumen donde se comparan los diferentes tipos de programas antes presentados.

*Tabla 1: Comparación entre los diferentes tipos de programas según su estructura*

Programa	Ventajas	Desventajas
Programas Lineales	<ul style="list-style-type: none"> <li>✓ Orden de la enseñanza establecido por el programador.</li> <li>✓ Enseñanza generalizada. Todos reciben el mismo conocimiento en la misma secuencia.</li> <li>✓ Fácil de programar.</li> </ul>	<ul style="list-style-type: none"> <li>✗ No contempla problemas individuales de cada alumno</li> <li>✗ El alumno puede no entender la explicación</li> </ul>
Programas Ramificados	<ul style="list-style-type: none"> <li>✓ Secuencia fija, contenido determinado por el programador.</li> <li>✓ Actúa según la respuesta del usuario gracias a la realimentación.</li> <li>✓ Posibilidad de repetir las explicaciones.</li> <li>✓ Fácil de programar.</li> </ul>	<ul style="list-style-type: none"> <li>✗ No contempla problemas individuales de cada alumno.</li> </ul>
Sistemas Generativos	<ul style="list-style-type: none"> <li>✓ Se individualizan las explicaciones a cada alumno.</li> <li>✓ El ordenador genera problemas a partir de las respuestas del individuo.</li> </ul>	<ul style="list-style-type: none"> <li>✗ Hay materias para las que no es válido por la dificultad de sus problemas.</li> <li>✗ Pueden existir varias soluciones para un mismo problema.</li> </ul>
ITS	<ul style="list-style-type: none"> <li>✓ Totalmente adaptada al alumno</li> <li>✓ Flexible y abierto a preguntas y modificaciones.</li> <li>✓ Enseñanza que imita al profesor.</li> </ul>	<ul style="list-style-type: none"> <li>✗ Estructura complicada.</li> <li>✗ Muy difícil de programar.</li> <li>✗ No válida en todos los casos.</li> </ul>

Para la creación de la aplicación de Mecánica de Máquinas se pretende una estructura sencilla y fácil de programar, para permitir la posterior creación de nuevos ejercicios. Es necesario que la aplicación se adapte a cada alumno, permitiéndole establecer la velocidad de aprendizaje. No se requiere el nivel de individualización que ofrecen los sistemas generativos y los ITS ya que la aplicación está pensada para ser un apoyo a la docencia y no un método docente único.

Por estos motivos, la estructura que se le dará a la aplicación de Mecánica de Máquinas será la de un programa ramificado con una secuencia fija para cada tipo de

problema a resolver (ver apartado 1.2, Cap.4) y con un sistema de retroalimentación que permita al alumno adecuar la velocidad de enseñanza a sus necesidades y repetir las explicaciones.

### 2.2.2) Programas EAO según su finalidad

En la actualidad existe una gran cantidad de programas y a pesar de ello, cada día se crean nuevos o se hacen modificaciones de los ya existentes. Los programas o “*software*” dedicados a la EAO se pueden clasificar en diferentes tipos según su finalidad [10]:

- Programas de productividad
- Programas de ejercicios de repetición
- Tutoriales
- Simuladores
- Herramientas de autor
- Sistemas integrados de aprendizaje

#### Programas de productividad

Este tipo de software basa su funcionamiento en conseguir un objetivo principal: trabajar más en menos tiempo. Para ello se deja al alumno libre para centrarse en las cuestiones que él considere más importantes y ayuda a facilitar y agilizar las tareas rutinarias y repetitivas.

Los programas de productividad facilitan el proceso de captura, manipulación, organización y presentación de los datos y la información. Todos estos procesos son en definitiva una experiencia de aprendizaje profunda ya que estimulan el descubrimiento y la construcción del conocimiento.

Dentro de los programas de productividad podemos encontrar varios tipos [10]:

- Procesadores de texto: (*Word*, *Open Office Word*, etc.). Son un tipo de herramienta que permite trabajar el desarrollo de habilidades comunicativas escritas y la asimilación de contenidos. Fomentan el aprendizaje mediante la experimentación y la prueba de sus opciones “ensayo y error”. Además, la escritura es una estrategia adecuada de aprendizaje (Squires 1984) que facilita la asimilación, adquisición y comprensión de conocimientos.

- Bases de datos: (*Access, Open Office Database, etc.*). Programas que permiten la introducción, almacenamiento, ordenación, clasificación, consulta, exportación y presentación de datos de cualquier tipo.
- Hojas de cálculo: (*Excel, Open Office Calc, etc.*). Trabajan con datos numéricos principalmente. La estructura de este tipo de programas suele ser una parrilla de celdas sobre las que se marcan las cifras y operaciones. Además permiten la representación de estos datos mediante gráficas.
- Infografía: (*Photo Shop, Gimp, Corel Draw, etc.*). Su objetivo es la creación de gráficos y dibujos de alta calidad incluso sin tener habilidades en la expresión gráfica. Fomentan la creatividad. Presentan gran cantidad de posibilidades: gráficas, históricos, creación artística, creaciones multimedia, etc.
- Autoedición y presentaciones: (*Power Point, Open Office Presentations, etc.*). Son un recurso muy utilizado en la enseñanza porque permiten crear presentaciones donde la información se presenta de una forma dinámica, concentrada y visual.

### Programas de ejercicios de repetición

Se basa en el fundamento “*Drill and Practice*” [10] (entrena y practica) para reforzar un conocimiento ya adquirido. Los ejercicios de repetición a veces pueden ser considerados una técnica importante de aprendizaje de conocimientos básicos y un refuerzo de habilidades intelectuales. Todo esto se realiza mediante ejercicios interactivos y una retroalimentación que según los resultados permitirá avanzar o hará repetir la lección.

Un ejemplo claro de este tipo de programas es la herramienta que se desarrolla en este proyecto. Se busca el aprendizaje de la asignatura mediante la realización de diversos ejercicios con el mismo patrón (ver apartado 1.1, Cap.4). Además se intenta dar un alto nivel de interactividad en el proceso de aprendizaje para llamar la atención del alumno y estimular así el avance y el progreso.

### Tutoriales

Son programas que enseñan al alumno nuevos conocimientos o habilidades con posibilidad de evaluar su comprensión. El programa avanzará o no, en función del resultado del usuario y a menudo van acompañados de ejercicios de repetición. Las principales características de un tutorial son:

- Exposición clara del objetivo que se quiere alcanzar.
- Incentiva la curiosidad del alumno.



- Da respuestas adecuadas y rápidas.

En la actualidad hay gran cantidad de ejemplos de tutoriales. Prácticamente todo el resto de programas y herramientas que aquí se explican poseen tutoriales que enseñan a utilizar la propia herramienta (sobre todo en programas complicados de gran versatilidad). Un ejemplo claro es la gran cantidad de tutoriales que existen para aprender a utilizar la herramienta “*Macromedia Flash*” que a su vez, como se explicará más adelante, se trata de una “herramienta de autor”.

## Simuladores

Introduce al alumno en una experiencia de ensayos sobre la realidad. La filosofía que sigue es la llamada constructivista que consiste en que el individuo experimenta y construye el conocimiento a partir de la experiencia.

Las principales características son: interactividad, realismo y respuestas rápidas. Actualmente hay muchos programas simuladores dedicados a la enseñanza, incluso estos se combinan con programas de diseño u otro tipo. Algunos ejemplos son los software de diseño y simulación como “*Abaqus CAE*”, “*Creo Parametric*”, “*Ansys Workbench*”, etc.

## Herramientas de autor

Son herramientas que permiten elaborar aplicaciones independientes del programa con el que se han creado. Permiten crear sistemas multimedia a la medida de las necesidades del autor.

Tienen un entorno de trabajo muy visual basado en iconos, objetos y menús de opciones. Trabajan con textos, imágenes videos audio y todo tipo de medios digitales y formatos de almacenamiento de información. Muchos de ellos incorporan un lenguaje de programación que les hace más versátiles y permiten al usuario crear programas más robustos y complejos. Algunos ejemplos son:

- *Silverlight* es un complemento de *Microsoft* que permite desarrollar aplicaciones para la web incorporando video, animaciones, interactividad e interfaces de usuario muy completas. Para ello utiliza gráficos vectoriales, archivos multimedia y una organización por capas. La animación se basa en una línea de tiempo y un marco de trabajo o “*Framework*”. Esto se complementa con un lenguaje de programación basado en XAML (el acceso a los objetos y eventos se realiza mediante *Visual Basic*, C y *Ajax*). Es compatible con los principales navegadores: IE, *Mozilla*, *Chrome*, *Opera*, *Safari*, etc. [11]

- *Tumult Hype* es una herramienta que permite crear animaciones para dispositivos iOS y para la web con una programación muy sencilla que no exige grandes capacidades. Es un software barato y versátil. La animación de objetos se basa en una línea de tiempo compuesta por “frames” o escenas. Su programación se basa en el lenguaje de HTML. [12]
- *Squeak* es un entorno de desarrollo integrado para elaborar aplicaciones gráficas interactivas cuya principal característica es la intuitividad. Reúne imágenes, vídeo, sonido, capacidad de simulación, editor, compilador, sistemas de ventanas, navegador, etc. Incluye un lenguaje de programación que evolucionó de *Smalltalk*. Se trata del primer lenguaje de programación orientado a objetos que permite concentrar al programador en la creación del sistema y no en el código y todo ello gracias a su intuitividad y a su semejanza con el lenguaje cotidiano. [13]
- *Macromedia Flash*: se trata de una herramienta de autor que utiliza gráficos vectoriales, gráficos rasterizados (imágenes), flujo de video y audio para la creación de animaciones. Está dedicado especialmente para la creación de aplicaciones web. Trabaja sobre capas, fotogramas y una línea de tiempo que se complementa con el lenguaje de programación llamado *ActionScript*. Es un lenguaje orientado a objetos y basado en XAML y *JavaScript* [14]. Se definirán más detalladamente las características de esta herramienta de autor en el apartado de *Macromedia Flash* (apartado 2.2.2.7, Cap.2)

En la Figura 4 se muestran por orden los logotipos de las herramientas de autor explicadas anteriormente.



*Figura 4: Ejemplo de Herramientas de autor. Silverlight, Tumult Hype, Squeak y Macromedia Flash (de izquierda a derecha)*

En el ámbito de la enseñanza, el caso de las herramientas o programas de autor se han desarrollado en otro sentido. El problema surge cuando los profesores necesitan crear una aplicación interactiva para la enseñanza, pero no tienen los conocimientos suficientes de programación. Se crean así los denominados “programas de autor”:

*“Los programas de autor son un tipo de aplicaciones de libre acceso que permiten a sus usuarios crear sus propios proyectos multimedia con poca o ninguna idea de programación” [15]*

Son programas cerrados, con limitaciones a cierta materia o tipo de ejercicios, menos versátiles pero muy intuitivos, fáciles de usar y sin ningún tipo de programación. Existen gran cantidad de ejemplos [15]:

- *Hot Potatoes* cuenta con seis aplicaciones para realizar actividades de opción múltiple, respuestas cortas, ordenar, asociar, rellenar huecos, etc.
- *JClic* para la creación de actividades educativas como ejercicios de texto, asociaciones, etc.
- *Exe Learning* es un editor de XHTML, permite crear cursos digitales completos, mapas de navegación web, permite empaquetar los contenidos como SCORM (ver apartado 2.4.4, Cap.2) y exportar a diferentes plataformas como *Moodle*.
- *EdiLim* es un editor de libros LIM, permite crear y visualizar ficheros xml (formato libros).
- *Lams* es una herramienta de autor para la creación de actividades colaborativas. Se pueden diseñar secuencias de contenidos digitales dirigidas a un gran grupo de individuos. Permite conocer el progreso del alumno, ver sus dificultades, cómo se enfrenta a la tarea, etc.
- *Malted* permite la creación de contenidos educativos digitales para idiomas



*Figura 5: Programas de autor dedicados a la enseñanza. Hot Potatoes, JClic, Exe Learning, Edilim, Lams, Mated (de izquierda a derecha) [15]*

## Análisis comparativo de las diferentes herramientas de autor

A continuación se justificará la elección del software más apropiado para la creación y el diseño de la aplicación de Mecánica de Máquinas. Las comparaciones se realizarán entre las diferentes herramientas de autor presentadas, es decir, en este apartado no se justifica de qué tipo es la aplicación de Mecánica de Máquinas, sino qué herramienta de autor se ha seleccionado para crearla.

Primero es necesario recalcar la diferencia entre las herramientas de autor y los programas de autor dedicados a la enseñanza, explicado en el apartado anterior. En la Tabla 2 se muestra una comparación de las características de ambas herramientas.

Tabla 2: Comparación entre las herramientas de autor y los programas de autor dedicados a la enseñanza

	Ventajas	Desventajas
Herramientas de autor	<ul style="list-style-type: none"> <li>✓ Gran versatilidad a la hora de crear aplicaciones.</li> <li>✓ Incluyen lenguaje de programación.</li> <li>✓ Creación de programas robustos.</li> <li>✓ Resultados reproducibles en la mayoría de navegadores y dispositivos (uso generalizado).</li> <li>✓ Disponibilidad de librerías y manuales de ayuda.</li> </ul>	<ul style="list-style-type: none"> <li>✗ Requiere tiempo para aprender su funcionamiento y todas sus posibilidades</li> <li>✗ Difícil programación.</li> </ul>
Programas de autor dedicados a la enseñanza	<ul style="list-style-type: none"> <li>✓ Uso muy intuitivo.</li> <li>✓ Funcionamiento y posibilidades fáciles de aprender.</li> <li>✓ Adecuados para la enseñanza y aprendizaje.</li> </ul>	<ul style="list-style-type: none"> <li>✗ No tienen lenguaje de programación.</li> <li>✗ Cerrados a la creación de un tipo concreto de aplicaciones.</li> <li>✗ Programas menos robustos.</li> <li>✗ Uso menos generalizado entre los desarrolladores.</li> </ul>

Vistas las ventajas y desventajas de los dos tipos de software, la elección para la creación de la aplicación de mecánica de máquinas será una herramienta de autor. Como se explica en el apartado de Análisis de Requisitos (apartado I, Cap.4) es necesaria una herramienta de autor que permita la creación de una aplicación dinámica, con una navegación robusta y que se pueda incorporar en la web, entre otros requisitos.

En segundo lugar es necesario comparar y evaluar la mejor elección dentro de las diferentes herramientas de autor. Según un informe de “*Stat Owl*” de 2012 [16], las más importantes eran *Adobe Flash*, *JavaFX*, y *Microsoft Silverlight*, con una penetración en el mercado del 96%, 76%, and 66% respectivamente.

A partir de septiembre del 2012 se aprecia una fuerte tendencia de las herramientas basadas en HTML5 en lugar de las herramientas basadas en *plug-ins*. Entre ellas destacan: *Java applet* 4.20, *Tumult Hype*, *ZK* 6.5, *GWT* 2.10, *Silverlight* 1.70, *Adobe Flex* 1.00 y *JavaFX* 0.5 entre otras.

HTML5 permite la creación de las conocidas “*Rich Internet Applications*”, pero el rendimiento que ofrecen no será el mismo que el que pueda ofrecer una aplicación desarrollada en *Flash/Flex*. HTML5 aún no es un lenguaje maduro y se encuentra en fase de desarrollo.

En contra partida, Flash tiene dos inconvenientes. El primero es la compatibilidad para reproducir los archivos generados. Es necesario disponer del reproductor *Flash* (si se ejecuta desde un PC o un dispositivo móvil) o de un navegador con el “*plugin*” especial de Macromedia instalado (si se reproduce en la web). Tanto el reproductor como el “*plugin*” son gratuitos y la mayoría de dispositivos y navegadores ya lo traen instalado de serie. El segundo

inconveniente es el tamaño de los archivos y el tiempo de carga. Sin embargo en el caso que se está estudiando, se creará un archivo por ejercicio, por lo que su tamaño no será excesivo.

Con estas justificaciones, la elección final para el desarrollo de la aplicación de Mecánica de Máquinas será Adobe Flash, software que se definirá con más detalle en el siguiente apartado.

## Macromedia Flash

Flash Mx es una de las más potentes herramientas de autor del mercado, creada por Macromedia, para el diseño de páginas web, animaciones, aplicaciones etc. Gracias a su gran versatilidad, su uso abarca multitud de campos como por ejemplo la educación (ODEAS, ver apartado 2.4.4, Cap.2), cine (dibujos animados, animación en películas), industria (optimización de planos, diseño de interiores), web (páginas web, blogs, banners publicitarios), etc.

Existen diversos manuales de utilización, tutoriales y bibliografía sobre esta herramienta, por lo que no es necesario hacer un análisis exhaustivo de todas las características y funcionalidades que incorpora. Para la creación de la aplicación de Mecánica de Máquinas se han utilizado unas herramientas concretas. Se definirán únicamente estas a continuación.

Flash se presenta al usuario mediante una interfaz dividida en diferentes secciones intercambiables para que el usuario organice su entorno de trabajo (Figura 6).

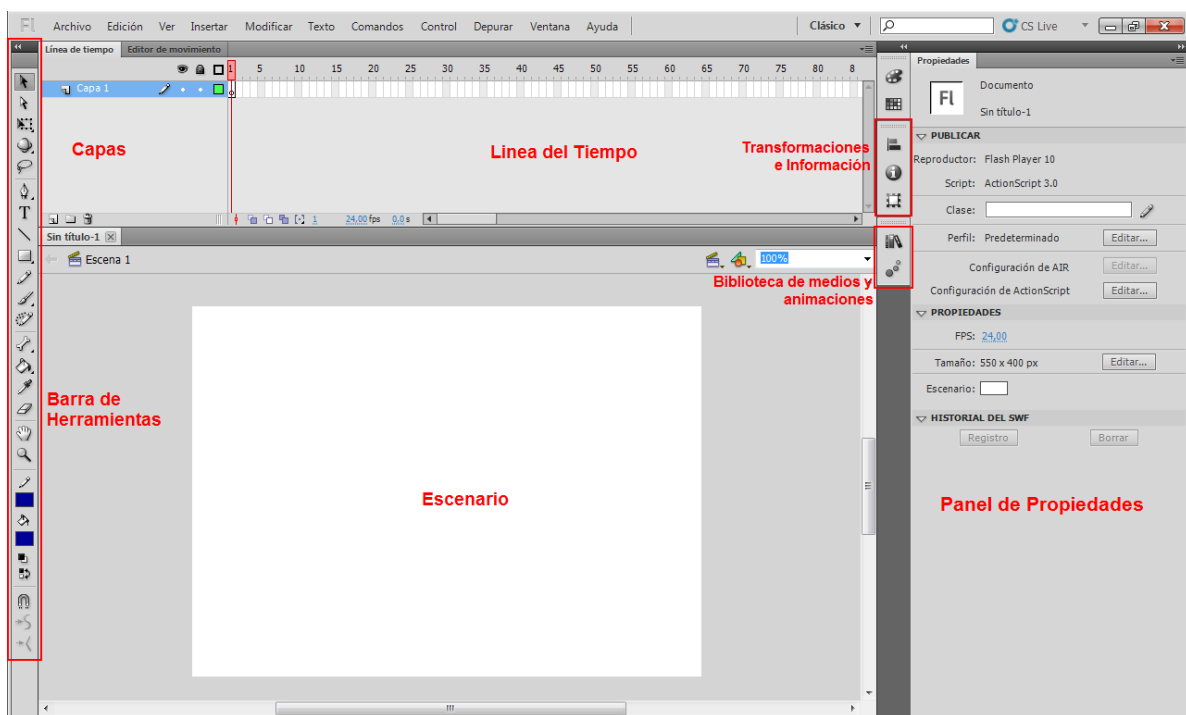






Figura 6: Interfaz de usuario de Adobe Flash Professional CS5

En la organización que viene por defecto, se diferencian las siguientes partes:

- El “Escenario” corresponde a la parte que se visualizará cuando se reproduzca el archivo. Se pueden crear elementos fuera de la región blanca, pero estos no se verán. Las principales propiedades del escenario (que aparecen en el “Panel de propiedades” al seleccionarlo) son sus dimensiones y la velocidad de reproducción de sus fotogramas, establecida a 24 FPS (fotogramas por segundo).
- El listado de las “Capas” usadas en la animación. El concepto de capas es fundamental para el funcionamiento de Flash. Cada capa es un nivel donde se pueden dibujar o insertar objetos, textos, sonidos. Cada una tiene su propia línea de tiempo. Todos los objetos de una misma capa se mezclarán, y serán independientes del resto de la capas. Además se pueden ordenar en carpetas y aparecerán en el “Escenario” por encima o por debajo del resto dependiendo de la posición que ocupen en el listado de capas. Existen varios tipos de capas [14]:
  -  Capas normales: Son las capas por defecto de Flash que cumplen todo lo anteriormente descrito.
  -  Capas guía: Se emplean en la animación de objetos para marcar su trayectoria. Se asocia a otra capa llamada “Capa guiada”, donde aparecerá el/los objetos que serán guiados.
  -  Capas máscara: Se usan conjuntamente con las capas enmascaradas. Son un tipo de filtros que solo dejan ver, de la capa enmascarada, lo que se dibuje en la capa máscara.
  -  Capas enmascaradas: Va asociada a una capa máscara. De esta capa solo será visible lo que tapa la capa máscara que se coloco por encima.



*Figura 7: Ejemplo del resultado de Capa máscara + Capa enmascarada [14]*

- La “Línea del tiempo” es el elemento fundamental para la animación. Representa la sucesión de fotogramas en el tiempo. Al igual que cualquier película, la animación en Flash se basa en la representación de multitud de fotogramas a una velocidad determinada.
- La “Barra de herramientas” donde se muestra al usuario todas las herramientas ordenadas por campos. Primero aparecen las herramientas de selección y transformación. A continuación las de creación de objetos, líneas y texto. Le

siguen las herramientas de color relleno y borrador. Por último aparece una serie de herramientas de zoom, posición y opciones de color. La función de cada herramienta se puede encontrar bien definida en los tutoriales de Flash o en la referencia [14].

- El “Panel de propiedades” donde se muestra toda la información y los parámetros referentes a la herramienta que se esté utilizando. No es un panel fijo ya que cada herramienta tiene unas propiedades diferentes.
- La “Barra de transformaciones” e Información presenta las propiedades del objeto en cuanto a posición, alineación con otros objetos, tamaño, giro, simetría, etc. El usuario puede cambiar estos parámetros de una forma más precisa que con las herramientas de selección.
- La “Biblioteca de medios y animaciones”, reúne todos los gráficos, clip de películas y botones utilizados. En Flash adquiere gran importancia el concepto de “Símbolo” que reúne a: gráficos, clip de película y botones (Figura 50). Cada uno tiene propiedades específicas y son esenciales para la creación de grandes animaciones ya que de un mismo símbolo se pueden hacer instancias (copias) disminuyendo mucho el tamaño de archivo final. Además, cambiando el símbolo original, todas sus instancias cambian. Se explicarán en detalle los símbolos utilizados para la aplicación de Mecánica de Máquinas en el apartado de diseño de la navegación (apartado 2.3.2, Cap.4).

Una de las características por las que se seleccionó esta herramienta de autor, es su capacidad para animar objetos y lo fácil que ello resulta. En la línea de tiempo aparecerán los denominados “Fotogramas claves” que son aquellos que tienen la información introducida por el usuario. Flash automáticamente rellena el resto de fotogramas posteriores con la información del fotograma clave. Así una animación puede hacerse de dos formas: fotograma a fotograma, introduciendo información diferente en cada uno de ellos; Mediante la “Interpolación” donde el usuario solo introduce información en el primer fotograma y en el último, encargándose el programa de crear el resto de fotogramas entre medias para que la animación sea progresiva. A su vez, Flash puede crear tres tipos diferentes de interpolación dependiendo de lo que el usuario necesite: interpolación de movimiento, de forma o la clásica. El único requisito es que el objeto animado sea un símbolo. Toda la información referente a la animación de objetos se puede encontrar en los tutoriales de Flash, en la bibliografía existente o en la referencia [14].

Otra característica relevante que llevó a la selección de Flash, es su lenguaje de programación “*Action Script 3.0*”. Se trata, como su nombre indica de un lenguaje de “*scripts*”, es decir, que no es necesario crear un programa completo, sino que, mediante órdenes aisladas, se puede conseguir los objetivos deseados.

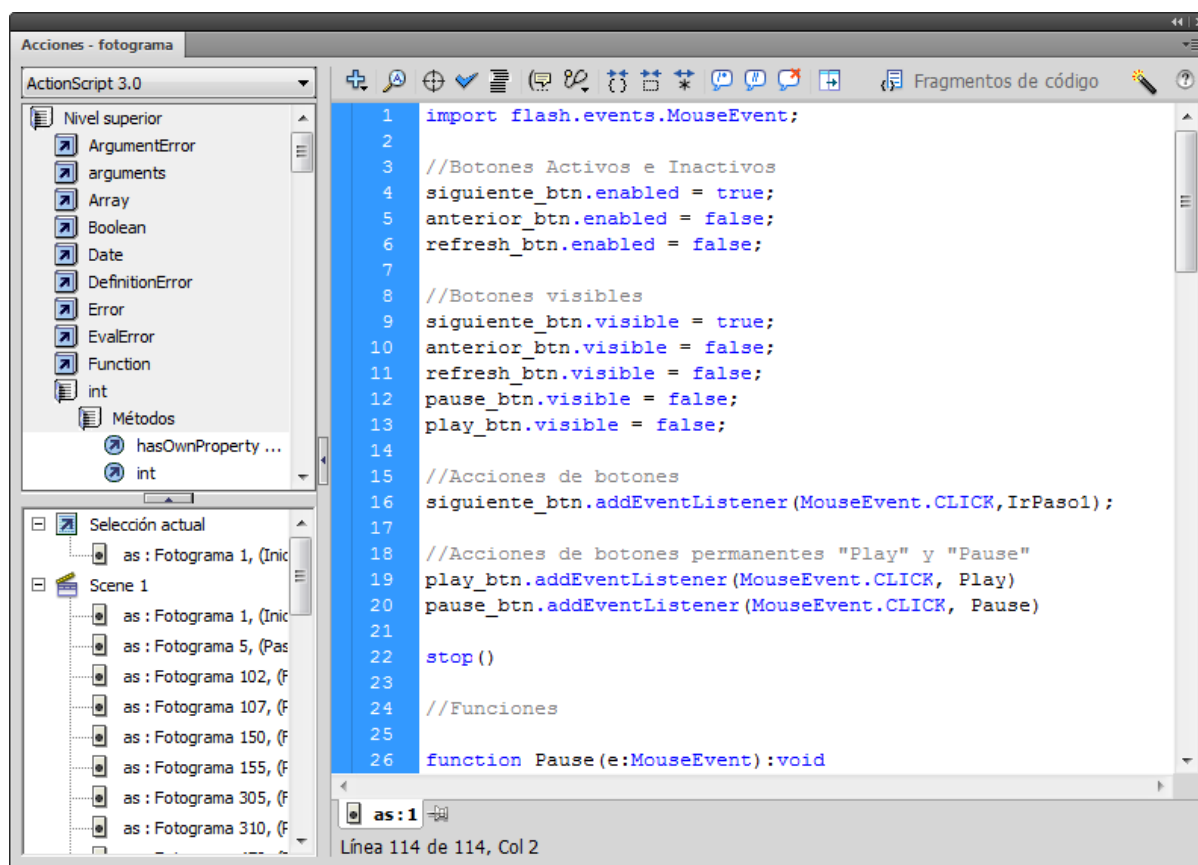
Action Script se trata de un lenguaje de programación orientado a objetos. Los objetos en sí se definen como:



*“En los términos más sencillos posibles, podemos pensar en un objeto como un conjunto de variables, constantes y funciones relacionadas, agrupadas juntas para crear un claro y completo bloque que podemos usar en nuestros script.” [17]*

Este lenguaje no tiene la potencia en cuanto a resultados y capacidades de un lenguaje puramente orientado a objetos derivados del C o *Pascal*, como por ejemplo *Visual Basic*. Sin embargo, la programación en Action Script es más sencilla y dispone de una colección de “funciones” (fragmentos de código) dispuestas al usuario para facilitar la programación.

Cada fotograma clave dispone de una ventana de “Acciones” (Figura 8) para insertar el código, que será ejecutado junto con dicho fotograma. Esta ventana a su vez está dividida en tres paneles, uno principal donde se escribe el código y dos a la izquierda donde aparecen una guía de acciones predeterminadas y un listado de todos los fotogramas que tienen código escrito.



**Figura 8: Ventana de “Acciones”. Panel de código (Derecha), Panel de Acciones (Izquierda superior), Listado de fotogramas con código (Izquierda inferior).**

A parte de todos los operadores que posee Action Script (como el resto de lenguajes), incorpora estas “Acciones” previamente comentadas, que son funciones predeterminadas, listas para usar, que agilizan y facilitan mucho la programación. Algunos ejemplos son: “*gotoAndPlay*” para enviar la cabeza lectora a un fotograma y reproducir; “*loadMovie*”



permite cargar otras películas o imágenes de forma dinámica; “play/stop” para reproducir o parar la película, etc.

Poder insertar las animaciones creadas en una página web, es otra de las características de esta herramienta de autor por la que fue seleccionada. Adobe Flash Professional CS5 puede crear dos tipos de archivos: “.swf” y “.html”. El primero es el formato de salida estándar. Para visualizar este tipo de archivos es necesario disponer del reproductor Flash. El segundo tipo es para insertar el archivo en una página web cuyo lenguaje de programación sea del estilo “html”. Para ambos archivos, el usuario dispone de diversas opciones de publicación de la película.

## 2.3) Enseñanza Virtual (e-learning)

El “*e-learning*” o enseñanza virtual es un método de enseñanza en el que se unen los beneficios de la enseñanza asistida por ordenador (EAO) y el uso de internet. Todas las formas de “*e-learning*” tienen como factor común la utilización de la red como tecnología de distribución de la información [19].

Se perfila como solución a los problemas a los que la enseñanza tradicional no da respuesta. El “*e-learning*” permite la utilización de herramientas con las que se consigue un aprendizaje más efectivo. A continuación se muestran las características principales de la enseñanza virtual (formación basada en red) y de la tradicional (formación presencial) para ver sus diferencias [20]:

**Tabla 3: Características de la formación presencial y en red (Cabrero et al., 2005) [20]**

Formación basada en la red	Formación presencial tradicional
<ul style="list-style-type: none"> <li>– Permite que los estudiantes vayan a su propio ritmo de aprendizaje</li> <li>– Es una formación basada en el concepto de <i>formación en el momento en que se necesita (just-in-time training)</i></li> <li>– Permite la combinación de diferentes materiales (auditivos, visuales y audiovisuales)</li> <li>– Con una sola aplicación puede atenderse a un mayor número de estudiantes</li> <li>– El conocimiento es un proceso activo de construcción</li> <li>– Tiende a reducir el tiempo de formación de las personas</li> <li>– Tiende a ser interactiva, tanto entre los participantes en el proceso (profesor y estudiantes) como con los contenidos</li> <li>– Tiende a realizarse de forma individual, sin que ello signifique la renuncia a la realización de propuestas colaborativas</li> <li>– Puede utilizarse en el lugar de trabajo y en el tiempo disponible por parte del estudiante</li> <li>– Es flexible</li> <li>– Tenemos poca experiencia en su uso</li> <li>– No siempre disponemos de los recursos estructurales y organizativos para su puesta en funcionamiento</li> </ul>	<ul style="list-style-type: none"> <li>– Parte de una base de conocimiento, y el estudiante debe ajustarse a ella</li> <li>– Los profesores determinan cuándo y cómo los estudiantes recibirán los materiales formativos</li> <li>– Parte de la base de que el sujeto recibe pasivamente el conocimiento para generar actitudes innovadoras, críticas e investigadoras</li> <li>– Tiende a apoyarse en materiales impresos y en el profesor como fuente de presentación y estructuración de la información</li> <li>– Tiende a un modelo lineal de comunicación</li> <li>– La comunicación se desarrolla básicamente entre el profesor y el estudiante</li> <li>– La enseñanza se desarrolla de forma preferentemente grupal</li> <li>– Puede prepararse para desarrollarse en un tiempo y en un lugar</li> <li>– Se desarrolla en un tiempo fijo y en aulas específicas</li> <li>– Tiende a la rigidez temporal</li> <li>– Tenemos mucha experiencia en su utilización</li> <li>– Disponemos de muchos recursos estructurales y organizativos para su puesta en funcionamiento</li> </ul>

Actualmente, existe un gran auge de los denominados MOOCs (*Massive Open Online Course*) que se trata de cursos a gran escala, dirigidos a un grupo grande de alumnos y de acceso libre a través de internet. Surgieron en el 2008 con el que sería el primer MOOC: “*Conectivisim and Connective Knowledge*” (G. Siemens y S. Downes) [21] y se ha extendido hasta un nivel que replantea el futuro de la educación superior y el papel de las universidades. Sus principales características son el número ilimitado de matrículas, el seguimiento online y el carácter abierto y gratuito. Hay gran cantidad de ejemplos de páginas web que ofrecen este tipo de cursos online:

- “*Miríada x*” [22] Es una plataforma Española impulsada por Telefónica y Universia que ofrece gran diversidad de cursos completos en los que se programan plazos para aprender sus diferentes módulos y para la entrega de trabajos, proyectos y exámenes. Está asociada con más de mil universidades de Iberoamérica y España, entre ellas la U. Carlos III.
- “*Coursera*” [23] Implementa videos, exámenes, interacción con otros estudiantes para aprender ciencia, nutrición, matemáticas, incluso medicina y hasta finanzas. Cuenta con más de medio millón de estudiantes de todo el planeta y los integra en una plataforma coherente y seria para el auto aprendizaje colaborativo.
- “*Khan Academy*” [24] se distingue por su variedad de temas y por la extensión de plataformas sobre las que funciona.
- “*Udacity*” [25] autodenominada la universidad del Siglo XXI es una plataforma para aprender resolviendo retos, con proyectos guiados con algunos de los mejores instructores. Posee más de cien mil estudiantes, básicamente de ciencias de la computación.
- “*edX*” [26] Institución sin ánimo de lucro creada por las universidades de Harvard y MIT (*Massachusetts Institute of Technology*) que ofertan MOOCs y clases online de gran variedad de asignaturas. Heredan los cursos del “*MIT Open Course Ware*”.

El sistema de “*e-learning*” también está presente dentro de la mayoría de las universidades españolas con los campus virtuales, aulas virtuales, etc. Responde ante las necesidades de los alumnos y eliminan las barreras que ocasiona la educación tradicional [21]:

- Problemas geográficos. Las personas que viven en zonas alejadas o no tienen posibilidad de acceder a la universidad; Trabajadores que son desplazados por sus empresas a diferentes lugares geográficos, lo que impide una continuidad en sus estudios; o a las personas discapacitadas que pueden tener problemas a la hora de desplazarse al centro de enseñanza.

- Problemas de tiempo. La enseñanza tradicional requiere establecer un horario en el que se reúnan profesores y alumnos. Sin embargo son muchas las personas que no pueden acudir al centro de enseñanza en ese horario.
- Problemas de demanda. Resulta complicado crear e impartir cursos formativos cuando la demanda local no justifica el esfuerzo ni la inversión. Las acciones formativas dirigidas a pequeñas poblaciones encuentran difícilmente justificación por escaso número de alumnos potenciales, aunque los mismos tengan una gran necesidad de formarse o reciclarse en determinadas materias.

Así la enseñanza virtual permite superar las barreras geográficas con una simple conexión a internet. Los horarios del alumno son flexibles adaptándose a sus necesidades y permitiendo la simultaneidad de estudios, trabajo o cualquier actividad. Los problemas de demanda quedan solventados tanto por parte del alumnado que puede encontrar gente de todo el mundo que esté interesada en la misma materia elevando la demanda; y por parte del profesorado al que se exige una menor inversión de dinero y de tiempo.

Sin embargo, presenta ciertos inconvenientes: una interactividad menor entre el profesor y el alumno; una retroalimentación lenta; es más difícil la rectificación de errores en los materiales, evaluaciones, etc.; hay más abandonos que en la enseñanza presencial; etc. [28].

Por todo esto, el desarrollo de las nuevas tecnologías intenta resolver estos problemas para conseguir que la enseñanza virtual sea cada vez más eficiente. Para ello se desarrollan programas con alto nivel de interactividad, herramientas de mensajería instantánea para comunicar profesor y alumno a tiempo real, cursos sencillos que estimulen la atención del alumno para evitar el abandono, etc.

La herramienta desarrollada para la asignatura de Mecánica de Máquinas en la que se centra este proyecto no está pensada para convertirse en una herramienta de enseñanza virtual completa. Se apuesta por la enseñanza semi-presencial en la que el alumno recibe los conocimientos teóricos de la forma tradicional de enseñanza y puede ponerlos en práctica gracias a esta herramienta de forma virtual desde cualquier lugar con conexión a internet.

## 2.4) La Web 2.0 y 3.0

En los últimos años de evolución de las nuevas tecnologías, Internet ha sufrido una revolución con la aparición de la Web 2.0, también conocida como la “web colaborativa”, “web social” o “web de las personas” [29]. Se puede definir como un conjunto de tecnologías para la creación social del conocimiento. Su principal característica es que incentiva la

creación colectiva de sus contenidos uniendo tres factores clave: tecnología, conocimiento y usuarios.

Posteriormente surge la Web 3.0 o “web semántica” [30] que parte de la base de añadir información adicional para complementar un contenido o incrementar el significado de los datos, logrando una mayor interoperabilidad. Esta web ha surgido del cambio en la forma de las relaciones sociales y comunicaciones entre las personas, como es el caso de las redes sociales

Son muy numerosas las apariciones de tecnologías colaborativas: *Blogs*, *Wikis*, *CMS*, *Redes sociales*, *feeds*, etc. Su uso también se ha llevado al ámbito de la educación (*Google docs*, *reader*, *academic*, *maps*, etc. *Moodle*, *Slideshare*, *Youtube*).

Todos estos nuevos medios de información y comunicación generan un contexto idóneo para el desarrollo de competencias tales como el pensamiento crítico, la autonomía, la iniciativa y el trabajo colaborativo entre otras muchas. Todas estas competencias son clave en el nuevo EEES.

El Parlamento Europeo y el Consejo (2006) establecieron ocho competencias clave para el aprendizaje en el nuevo EEES que tienen mucha relación con las que ahora generan las herramientas de la Web 2.0 [29]:

- Comunicación en la lengua materna.
- Comunicación en lenguas extranjeras.
- Competencia matemática y competencias básicas en ciencia y tecnología.
- Competencia digital.
- Aprender a aprender.
- Competencias interpersonales, interculturales y sociales, y competencia cívica.
- Espíritu de empresa.
- Expresión cultural.

La nueva filosofía de la Web 2.0 guarda mucha relación con casi todos los principios que se establecen para EEES y por lo tanto es un avance que encamina a los alumnos a conseguir los objetivos del Proceso de Bolonia. La Web 2.0 es en definitiva un modelo educacional constructivista que pone al alumno en el centro del proceso con un papel activo en su propio proceso de aprendizaje.

A continuación se exponen cuatro tipos de herramientas nacidas con la Web 2.0 que están emergiendo con gran fuerza y se están implementando en diferentes contextos: los portafolios digitales o e-portfolios; las redes sociales; los documentos y aplicaciones en la red

(*Cloud Computing*); y los Objetos Digitales de Enseñanza-Aprendizaje (ODEA). Además se analizará su repercusión en la enseñanza.

### 2.4.1) Los portafolios digitales (E-Portfolios)

Son una herramienta con la que un individuo recopila información y materiales bajo su iniciativa, para ser utilizados con diversos propósitos: como herramienta de aprendizaje; para la evaluación y acreditación de competencias; simplemente para compartir conocimiento.

Por un lado permite al individuo aprender a planificarse y autogestionar su propio proceso de aprendizaje. Consigue que el estudiante deje de ser un usuario pasivo para convertirse en un productor de información.

Los portafolios digitales se pueden clasificar en tres tipos [29]:

- De aprendizaje: herramienta de apoyo al proceso de enseñanza, diseñado y administrado por el estudiante y supervisado por el profesorado.
- De evaluación: una forma de aportar una dimensión cualitativa mediante la argumentación y la justificación de la adquisición de competencias.
- De presentación: recopilatorio de trabajos orientado a mostrar las competencias que ha adquirido el individuo.

El portafolio es, por tanto, una herramienta versátil con un amplio abanico de posibilidades. Actualmente se asocia a otras herramientas integradas en muchas universidades como: entornos virtuales de enseñanza-aprendizaje (como por ejemplo “*moodle*”); experiencias de materiales puestos a disposición de todo el mundo (“*Open Course Ware*”).

### 2.4.2) Las redes sociales

Las redes sociales han cambiado radicalmente la forma de comunicarse y compartir información de las personas. Suelen estar basadas en una plataforma web y ofrecen servicios de comunicación por correo electrónico o mensajería instantánea, blogs, foros, galerías multimedia y en definitiva permiten compartir cualquier tipo de archivo.

La definición que engloba a todas ellas (Boyd, 2007 [29]) es la de un medio que permite a los individuos crear un perfil público o semipúblico dentro de un sistema delimitado, crear una lista de otros usuarios con los que compartir su información, y ver la información de otros usuarios que se lo permitan.

Existen multitud de redes sociales que se crearon para su utilización en diferentes ámbitos: “*Facebook*” se creó para atender necesidades en las redes universitarias americanas y actualmente es una red social donde los usuarios comparten sus experiencias; “*Linkedin*” se define como una red profesional orientada a formar contactos de tipo laboral y comercial; “*Youtube*” nació siendo un depositario de archivos de video y se ha convertido en un lugar donde los usuarios comparten sus videos caseros y profesionales; “*Mendeley*” es una red de carácter profesional donde se intercambian documentación y opinión entre investigadores; “*Livemocha*” para el aprendizaje de idiomas de forma colaborativa; otras muchas redes sociales están emergiendo para cubrir las necesidades de los usuarios.

En cuanto al tema de la educación, muchas universidades han apostado por integrarse en redes sociales existentes o por la creación de nuevas redes sociales propias generando un nuevo escenario de colaboración, internacionalización y transferencia del conocimiento.

### 2.4.3) Documentos y aplicaciones en la red (*Cloud Computing*)

El término “*Cloud Computing*” conocido como “la nube” describe a un grupo de ordenadores conectados en red que ponen a disposición del usuario un conjunto de infraestructuras de aplicaciones, almacenamiento y procesamiento de datos.

Su objetivo principal es permitir al usuario tener acceso a sus documentos, correo, agenda, o cualquier tipo de datos desde cualquier ordenador o dispositivo con el único requerimiento de tener una conexión a internet.

Actualmente hay multitud de ejemplos que ofrecen estos servicios: “*Google*”, “*Yahoo*”, “*Dropbox*”, “*Flirck*”, “*Youtube*”. Existe una cantidad incalculable de aplicaciones y páginas web de este tipo, ya que cada día se crean nuevas, o se modifican existentes.

A nivel educativo, estos entornos generan un espacio idóneo para la construcción compartida del conocimiento. Son herramientas imprescindibles para compartir materiales y recursos y realizar de una forma colaborativa trabajos y experiencias de investigación, con independencia de la distancia y el tiempo. Las aplicaciones y servicios de “la nube” están provocando un cambio en la forma de utilizar el “*software*”. Seguramente estas herramientas sean cada vez más importantes en la educación del futuro.

#### 2.4.4) Objetos Digitales de Enseñanza-Aprendizaje (ODEA)

Un ODEA se trata de cualquier recurso online en formato digital con el objetivo final de enseñar. Abarcan un gran abanico de posibilidades, que crece cada vez más gracias al desarrollo de las TIC. Se pueden encontrar aplicaciones, sistemas interactivos, textos multimedia, videos, audios, etc.

Los objetos de aprendizaje, también llamados RLO (*Reusable Learning Objects*), surgieron con Wayne Hodgins (1992) [31] que llevó por primera vez el proceso de aprendizaje de un niño a las nuevas tecnologías de información.

*“Un objeto de aprendizaje (ODA) es una colección de recursos de información ensamblada usando metadatos, para corresponder a las necesidades y personalidad de un aprendiz en particular. Múltiples objetos de aprendizaje pueden ser agrupados en conjuntos más grandes y animados entre sí para formar una infinita variedad y tamaños.”* Wayne H. (1992). [31].

Posteriormente, muchas empresas dedicadas al sector del diseño software como Oracle o Cisco Systems tomaron la idea y comenzaron a desarrollarla. Así, a finales de los años 90, el concepto estaba totalmente asentado al ámbito de la enseñanza. Sin embargo, no existía una clasificación clara.

David Willey (2001) propone una nueva definición: *“Cualquier recurso digital que puede ser usado como soporte para el aprendizaje”*. [31]

Fue necesaria la creación de estándares que regulasen la creación de estos objetos y que establecieran características comunes a todos (algunos estándares: AICC, LOM, IMS, etc.). El estándar SCORM (2004) [32] (*Sharable Content Object Reference Model*), el más relevante, establece los *“métodos coherentes en materia de almacenamiento, de identificación, de condicionamiento de intercambios y de recuperación de contenidos”*. Cumpliendo estos estándares, se asegura un aprovechamiento del potencial y la usabilidad que pueden ofrecer estos objetos.

Actualmente, algunas de las características de los Objetos Digitales de Enseñanza-Aprendizaje son las siguientes [31]:

- Capacidad de actualización y modificación constante gracias a su formato digital.
- Accesibilidad a muchos usuarios simultáneamente
- Incorporación, no solo del contenido para aprender sino también el camino para conseguir un aprendizaje satisfactorio.
- Contenido interactivo que implica la participación activa del alumno. Para ello, incorporan actividades, simulaciones, gráficas, evaluaciones, etc.



- Mantienen al alumno cognitivamente activo, captando su atención y consiguiendo el enfoque que el creador de la aplicación desea conseguir.
- Es indivisible e independiente de otros ODEAs. Por este motivo tiene sentido por sí mismo, es autocontenido y no se puede descomponer.
- Es reutilizable en otros contextos educativos diferentes para el que fue creado.

La gran expansión de los objetos de aprendizaje se debe a las ventajas que presenta tanto para estudiantes como para profesores. Las más destacables son el ahorro de tiempo para el profesor y la disponibilidad constante para el alumno. En la siguiente tabla (Tabla 4) se muestran otras ventajas.

*Tabla 4: Ventajas de los objetos de aprendizaje. Fuente [31]*

VENTAJAS	ESTUDIANTES	PROFESORES
<b>Personalización</b> (Adaptación del temario y la planificación temporal a cada estudiante)	<ul style="list-style-type: none"> <li>■ Individualización del aprendizaje en función de sus intereses, necesidades y estilos de aprendizaje.</li> </ul>	<ul style="list-style-type: none"> <li>■ Ofrecen caminos de aprendizaje alternativos.</li> <li>■ Adaptan los programas formativos a las necesidades específicas de los estudiantes.</li> </ul>
<b>Interoperabilidad</b>	<ul style="list-style-type: none"> <li>■ Acceden a los objetos independientemente de la plataforma y hardware.</li> </ul>	<ul style="list-style-type: none"> <li>■ Utilizan materiales desarrollados en otros contextos y sistemas de aprendizaje.</li> </ul>
<b>Inmediatez/ accesibilidad</b>	<ul style="list-style-type: none"> <li>■ Tienen acceso, en cualquier momento, a los objetos de aprendizaje que se desee.</li> </ul>	<ul style="list-style-type: none"> <li>■ Obtienen, al momento, los objetos que necesitan para construir los módulos de aprendizaje.</li> </ul>
<b>Reutilización</b>	<ul style="list-style-type: none"> <li>■ Los materiales ya han sido utilizados con criterios de calidad.</li> </ul>	<ul style="list-style-type: none"> <li>■ Disminuyen el tiempo invertido en el desarrollo del material didáctico.</li> </ul>
<b>Flexibilidad</b>	<ul style="list-style-type: none"> <li>■ Se integran en el proceso de aprendizaje.</li> <li>■ Se adaptan al ritmo de aprendizaje del alumno.</li> </ul>	<ul style="list-style-type: none"> <li>■ Es de fácil adaptación a:               <ul style="list-style-type: none"> <li>- los distintos contextos de aprendizaje.</li> <li>- las diferentes metodologías de enseñanza-aprendizaje.</li> </ul> </li> </ul>
<b>Durabilidad/ Actualización</b>	<ul style="list-style-type: none"> <li>■ Acceden a contenidos que se adaptan fácilmente a los cambios tecnológicos.</li> </ul>	<ul style="list-style-type: none"> <li>■ Crean contenidos que pueden ser rediseñados y adaptados a las nuevas tecnologías.</li> </ul>



Gracias a la característica de interoperabilidad que poseen los objetos, surge un nuevo concepto conocido como “objetos de acoplamiento” [33]. Se trata de objetos digitales que, unidos a otros objetos, tienen la función de contextualizar al alumno. Estos dan sentido a los objetos de aprendizaje.

En realidad los objetos de acoplamiento no cumplen por si solos las características de un ODEA. Dependen totalmente del objeto de aprendizaje al que acompañan y enlazan las distintas partes y elementos que componen el módulo de aprendizaje.

Los tipos de objetos de acoplamiento son los siguientes:

- Guía introductoria: Orienta al alumno y le muestra los posibles caminos de aprendizaje. Ofrece una visión general.
- Actividad práctica: Tareas estructuradas que pueden intercalarse a lo largo del módulo y donde el alumno puede demostrar su progreso.
- Evaluación: Ofrecen una realimentación al estudiante de su propio aprendizaje y detectan dificultades que pueda tener.
- Resumen: Donde se sintetizan las ideas que se presentan en el módulo. Su objetivo es facilitar la retención de lo aprendido y favorecer futuros aprendizajes.

En definitiva una ODEA es un recurso interactivo sencillo y versátil que unido con otros objetos digitales generan herramientas de aprendizaje efectivas tanto para el alumno como para el profesor.

# *CAPÍTULO 3*

## **DISEÑO DE UNA APLICACIÓN INTERACTIVA**

---

# I) LA APLICACIÓN Y SU INTERFAZ DE USUARIO

## 1.1) Ciclo de vida de una Aplicación Interactiva

La creación de una aplicación interactiva es un proceso largo y en constante movimiento. Se define como un ciclo continuo, en el que se distinguen varios estados: análisis de requisitos, diseño, implementación, lanzamiento, prototipado y evaluación. Por otro lado, durante el ciclo de vida de un sistema interactivo, adquieren gran importancia tres elementos: diseñador, aplicación y usuario. Todos ellos participan activamente a lo largo de todo el proceso, desarrollando una función específica en cada estado.

Las primeras aplicaciones, se creaban bajo las limitaciones de los computadores, aún en desarrollo. Con el crecimiento de las nuevas tecnologías, se busca que los sistemas interactivos estén diseñados por y para el usuario, de tal forma que recaiga sobre este toda la importancia del ciclo de vida de la aplicación. Un sistema interactivo será más eficaz, cuanto mejor acogida tenga entre los usuarios. Este será por lo tanto, el que tenga la última palabra en el proceso de creación de una aplicación

El ciclo de vida de un sistema interactivo [34] comienza con el **análisis de requisitos**. En esta primera etapa se determinará el cauce de la aplicación. Se fijarán los requisitos y los objetivos que se espera conseguir. La calidad del software se alcanzará cuando se llegue a estos requisitos. Por otro lado, es necesario aplicar técnicas de análisis para obtener las necesidades y las experiencias de los usuarios. Las principales actividades que hay que llevar a cabo son las siguientes:

- Análisis etnográfico, es decir, observar cómo trabaja el usuario y obtener así la forma más eficaz de acercarles la aplicación.
- Realizar un estudio del perfil del usuario que utilizará la aplicación, sus conocimientos y la tecnología de que dispone. Es importante tener en cuenta los roles del entorno donde funcionará el sistema a estudio.
- Establecer los objetivos funcionales, de usabilidad y de accesibilidad de la aplicación.

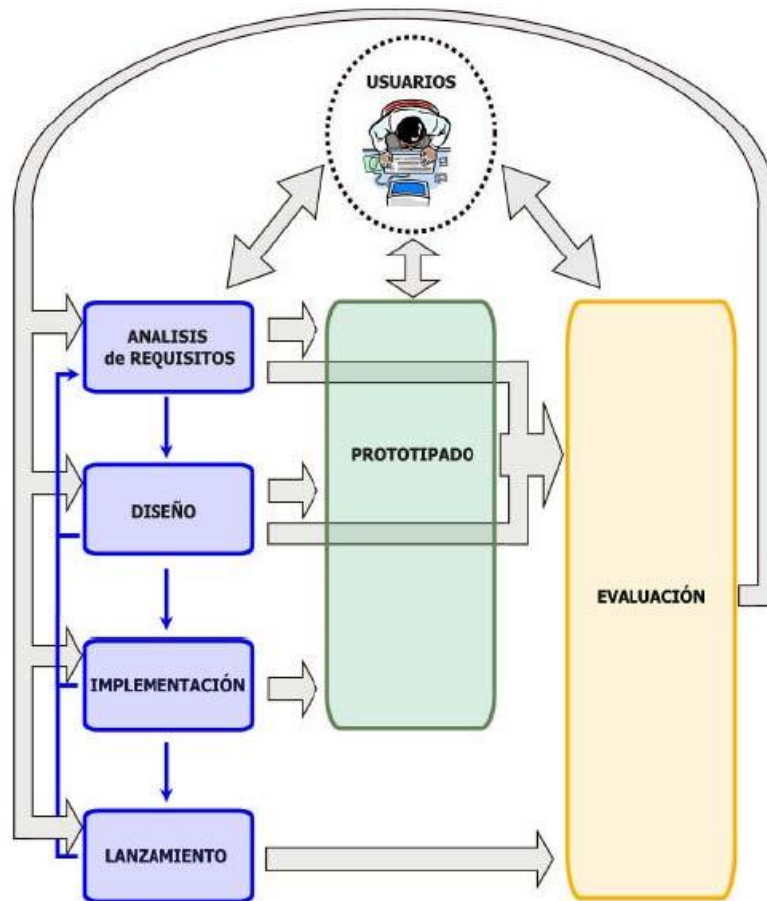
La siguiente etapa corresponde con el **diseño** de la aplicación. El diseñador ha de traducir todos los estudios realizados en la etapa anterior, al lenguaje del computador. Para ello no basta solo con crear un buen modelo funcional, sino que hay que tener presente el modelo mental del usuario (ver apartado 2.3, Cap. 3). En otras palabras, es más importante diseñar una aplicación intuitiva, que una aplicación muy versátil pero compleja. Para ello, es necesario llevar a cabo las siguientes actividades:

- Análisis del funcionamiento de la aplicación, es decir, las tareas que va a realizar y cómo las va a realizar. Es necesaria una descripción sistemática de las interacciones usuario-sistema.
- Analizar el entorno donde se enmarcará la aplicación (Sistema Operativo, web, etc.) y recopilar los estándares, guías de estilo, principios y recomendaciones relacionados con este entorno (ver apartado 3, Cap. 3).
- Definir el estilo de la aplicación, su presentación al usuario (metáforas, diseño gráfico, iconos, ventanas, etc.)

Después de haber llevado a cabo todo el proceso de análisis, comienza la etapa de la **implementación**. En esta fase, el papel del diseñador adquiere gran relevancia, ya que tiene que poner en práctica sus conocimientos y sus recursos informáticos personales. Aún así, hay dos factores que minimizan este esfuerzo: un trabajo satisfactorio en las etapas previas y la utilización de herramientas para el desarrollo de interfaces modeladas. Estas herramientas suponen una ayuda al diseñador, a la hora de implementar técnicas de interacción o presentar la información. Un ejemplo son las librerías incluidas en los lenguajes de programación, o los programas de autor, que incorporan módulos de diseño gráfico asistido (librerías de *Java*, asistente de programación de *AcciónScript*, biblioteca de botones y animaciones de *Macromedia Flash*, etc.).

Con la aplicación ya desarrollada, llega el momento del **lanzamiento**. En esta etapa es importante que el usuario se sienta cómodo con el sistema y que este sea accesible y usable. Sin embargo, es necesario mantener la realimentación del usuario de tal forma que el proceso se convierta en un ciclo. Por este motivo, el elemento más importante en esta fase es el usuario que decidirá si la aplicación se adecúa a sus expectativas o no.

Hasta ahora, el proceso seguido para la creación de un sistema interactivo se ha descrito como un flujo lineal. Sin embargo como vemos en la siguiente figura (Figura 9) el flujo se ramifica apareciendo dos fases paralelas al proceso: el prototipado y la evaluación.



*Figura 9: Ciclo de vida de un Sistema Interactivo*

La etapa del **prototipado**, se encuentra paralela al flujo de creación, porque no posee un lugar fijo dentro de este. El motivo es que desde el inicio del desarrollo es necesario crear y probar partes del sistema. La función del prototipo es:

- Verificar que el funcionamiento es el correcto y que se cumplen los objetivos establecidos.
- Averiguar aspectos relacionados con la interfaz del sistema (posición de los controles, textos, colores).
- Validar la navegación a través de la aplicación y la interacción con la misma.
- Realizar ensayos de mejoras, nuevas posibilidades técnicas, etc.

En la herramienta sometida a estudio por este proyecto, es muy importante la creación de un prototipo final que se asemeje lo máximo posible al resultado último de la aplicación. Por este motivo se explicará de forma más exhaustiva los principios del prototipado en el apartado de prototipos (apartado 1.5, Cap. 3) y los prototipos creados para la aplicación (apartado 3, Cap. 4).

Volviendo al flujo para la creación de un sistema interactivo, se observa también en paralelo, el proceso de **evaluación**. Se trata de un conjunto de metodologías y técnicas para el análisis del funcionamiento, usabilidad y accesibilidad del sistema. Al igual que el proceso de prototipado, la fase de evaluación no se realiza en un punto concreto del flujo (con la aplicación ya acabada como cabría esperar), sino que se han de evaluar cada una de las partes a lo largo del ciclo de vida de la aplicación. Se considera uno de los puntos clave en el proceso ya que permite recibir realimentación de los usuarios y mejorar así su experiencia. Se definirán posteriormente en el apartado de evaluación (apartado 1.6, Cap. 3), las técnicas referentes a esta etapa y se realizará un estudio de la aplicación a modo de evaluación final (apartado 4, Cap. 4).

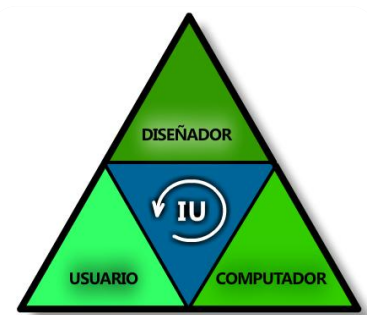
El ciclo de vida de un sistema interactivo no acaba con la evaluación del producto y la conformidad del usuario. Es importante mantener la realimentación e incorporar nuevas mejoras que adapten la aplicación al incesante desarrollo de las tecnologías. Se puede decir que, para que una aplicación perdure, ha de estar en constante evolución. Esta es la filosofía de muchos de los sistemas más comercializados en la actualidad. Mantienen la fidelización de sus usuarios mediante la continua actualización de sus productos, la realización de encuestas de satisfacción, o mediante la adaptación de sus sistemas a otros entornos (por ejemplo, el gran auge de las aplicaciones para terminales móviles). Hay que recordar que una aplicación tendrá éxito o no, dependiendo de la aceptación entre el público al que va dirigida.

La información de este apartado está basada en el estándar ISO TR 18529 (descripciones de los procesos del ciclo de vida centrados en la persona) que se explica más detalladamente, junto con otros estándares y normas (apartado 3, Cap.3).

## 1.2) Interfaz de Usuario

Todo sistema interactivo se caracteriza por la capacidad de establecer un diálogo entre la máquina y el usuario. En este diálogo, la interfaz de usuario (IU) es el punto de contacto entre estos dos elementos. Por este motivo se considera una parte fundamental en el diseño de una aplicación interactiva

Todo sistema está formado por tres componentes: el usuario, que posee la capacidad de elección y actuación; el ordenador, que ofrece la aplicación al usuario y los mecanismos para su uso; el diseñador, que se anticipa a las acciones del usuario y las programa para realizar la aplicación. Estos tres elementos se relacionan entre sí a través de la interfaz de usuario.



*Figura 10: Elementos de un Sistema Interactivo*

La interfaz puede ser un elemento “hardware” o “software” que facilite al usuario el acceso a los recursos del ordenador. En este caso se estudiarán únicamente los componentes “software” ya que, para la herramienta que se está diseñando, no se requieren componentes “hardware” que estén en contacto con el usuario, más que el propio ordenador.

Como sugirió *Thimbleby (1990)* [34] la interfaz determinará en gran medida la percepción y la impresión que el usuario poseerá de la aplicación. El usuario no busca conocer la estructura interna de la aplicación, sino cómo se usa. Así, en el proceso de creación de un sistema interactivo, el diseño de la interfaz no debe ser la etapa final. Debe tenerse en cuenta desde el principio, durante la creación del sistema y al finalizar. De otra forma, se obtendrá una interfaz muy dependiente de la estructura del sistema y su programación.

Un sistema interactivo debe funcionar por y para el usuario. Un punto a tener en cuenta a la hora del diseño, es que los usuarios tendrán distintos niveles de conocimientos. Por ello, no existe una interfaz válida para todos, sino que es necesaria cierta flexibilidad y la existencia de varias formas de interacción.

En el desarrollo de aplicaciones interactivas será necesario aplicar las técnicas de la ingeniería de software, así como la normativa dirigida al diseño de interfaces amigables.

## 1.3) Evolución de las Interfaces de Usuario.

La historia de las IU ha pasado por cuatro ciclos durante los diez últimos años. Comenzó con las Interfaces de Texto, posteriormente con las Interfaces de Menús, seguidas de las Interfaces Gráficas de Usuario (GUI) hasta llegar a la Interfaz Web.

Las antiguas IU requerían que el individuo trabajase bajo las condiciones del ordenador. Era necesario especificar cada acción o subrutina para que el ordenador la ejecutase. Se denominaban Interfaces de Texto o Interfaces de Línea de Mandatos (*Command\_line User Interface*, CUI). Poseían un lenguaje formal con vocabulario y sintaxis propia. No era intuitivo y requería un conocimiento previo (ejemplo, Interfaz *MS-DOS*, Figura 11).



Figura 11: Consola de Windows basada en Interfaz de Línea de Mandatos (CUI)





**Figura 12: Interfaz de Menú de Windows**

Más tarde se introdujeron los iconos y gráficas. Además se desarrollaron los dispositivos de entrada hardware (teclado y ratón entre otros). Surgen las Interfaces de Menús que permiten: navegar dentro de un sistema y seleccionar elementos de una lista que representa acciones (Figura 12).

Posteriormente se empezarán a utilizar metáforas en el diseño (aspecto de escritorio) naciendo el concepto de “GUI”. Este tipo de interfaz fue diseñado originalmente por Xerox (*Xerox Star*, 1981) y extendido por Apple (*Apple Macintosh*, 1984). Se caracterizaba por permitir manipular la información en la pantalla, no solo presentarla. Utilizan el estilo objeto-acción, a diferencia del estilo acción-objeto que utilizaban las Interfaces de Menús (Figura 13).



**Figura 13: Evolución en la interfaz de Windows. Escritorio de Microsoft Windows XP (atrás), Microsoft Windows 7 (delante.)**

La definición de Interfaz Gráfica de Usuario se ha ampliado considerablemente con la aparición de la Web 2.0. En un primer momento eran menús e iconos puramente funcionales. Actualmente las interfaces Web incorporan a su vez más interfaces, basadas en todo tipo de metáforas (documentos impresos, tiendas online, clases para la enseñanza, etc.) (Figura 14).



Con el avance en el diseño de nuevas interfaces de usuario se aprecia una tendencia en conseguir que estas sean cada vez más fáciles e intuitivas. Por este motivo, su creación es más compleja y requiere una mayor profundización en su apariencia. Esto ha obligado a los diseñadores a establecer una serie de convenios y directrices que regulen, en la medida de lo posible, las decisiones correctas para diseñar una interfaz efectiva. Sin embargo la incesante evolución de las tecnologías, y el impredecible comportamiento de los usuarios, hace que las IU y su diseño estén en un constante cambio.



Figura 14: Interfaz Web. Metáfora de una tienda

## 1.4) Estrategia de diseño de una IU

Para diseñar correctamente una IU, hay que tener en cuenta su estructura. Esta se asemeja, por su similitud, a la forma de un “iceberg” donde lo realmente visible para el usuario, la punta, es un porcentaje mucho menor que lo que conlleva la creación total de una IU eficiente. [35].

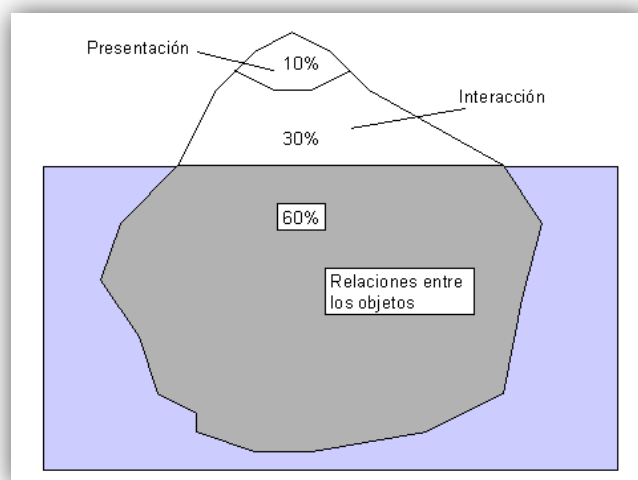


Figura 15: Estructura de una IU: “look-and-feel iceberg” de IBM (1992)

La parte importante es la que hay por debajo: los mecanismos de interacción, y la gestión de las entradas que se producen cuando el usuario interactúa con la aplicación (lo que se conoce como comunicación entre objetos).

En este apartado se estudiarán las tres partes de las que se compone una IU, atendiendo al siguiente orden:

- Mecanismos básicos de interacción.
- Gestión de las entradas del usuario (relaciones entre los objetos).
- Diseño gráfico de la interfaz (presentación).

### 1.4.1) Mecanismos de interacción

En el campo de la informática, la interacción del usuario con la aplicación, se traduce en la adquisición, por parte de la máquina, de unidades de información. Estas poseen un significado relacionado con dicha aplicación.

El diseño de las interacciones se basa en muchos conceptos y modelos de las disciplinas de ergonomía, semiótica, inteligencia artificial y ciencia cognitiva.

El tipo de interacciones que se pueden realizar son las siguientes [34]:

- Posicionamiento. Obtención de una posición u orientación. La posición puede ser en cualquier dimensión (2D, 3D). Se puede usar para dibujo, situar ventanas, arrastrar elementos, etc.
- Valor. Obtención de un dato cuantificable, ya sea numérico o porcentual.
- Texto. Introducción de un texto o identificador.
- Selección. Obtención de una alternativa entre varias posibles. Se puede distinguir entre la selección sobre un conjunto finito o bien de tamaño variable.
- Arrastre. Secuencia de posiciones entre un punto inicial y otro final (por ejemplo movimiento de una carpeta con el ratón).

Se estudiará cada uno de los mecanismos por separado para profundizar en los aspectos importantes relacionados con el diseño de una aplicación interactiva.

#### 1.4.1.1) Mecanismo de posicionamiento

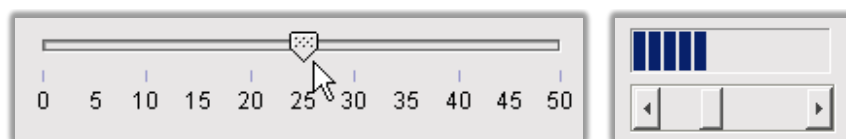
La información obtenida por la máquina consiste en una coordenada 2D o 3D. Esta acción se lleva a cabo generalmente con un hardware periférico (ratón, panel táctil, etc.) o mediante la introducción directa de la coordenada.

Algunos de los aspectos relacionados con el posicionamiento, que hay que tener en cuenta a la hora de diseñar una IU, son:

- Sistema de coordenadas: Es importante definir correctamente el sistema de coordenadas para conseguir una correcta orientación del usuario. Generalmente, en 2D, se utilizan los ejes de la pantalla (para referenciar la posición del cursor del ratón por ejemplo). Sin embargo, puede referenciarse el sistema de coordenadas a un objeto (por ejemplo, en aplicaciones de modelado en 3D).
- Resolución: En el caso de movimiento con un dispositivo (cursor) las posiciones pueden ser discretas o continuas. Habrá que tener en cuenta la relación Control/Display (C/D) para mecanismos de posicionamiento indirecto, es decir, la distancia que hay que mover el puntero (ratón) en la mesa para obtener una nueva posición en pantalla.
- Realimentación. Consiste en la información que recibe de vuelta el usuario al realizar una acción. La realimentación puede ser espacial (visualización del cursor en pantalla), o lingüística (representando el valor numérico en coordenadas cartesianas).

#### 1.4.1.2) Mecanismo de introducción de valor

Consiste en la obtención de un dato cuantificable introducido por el usuario. La forma más habitual, en el caso de requerir un valor numérico, es la entrada mediante teclado. Sin embargo, se pueden encontrar representaciones gráficas (diales, deslizadores, etc.) (Figura 16) que permiten la introducción, mediante posicionamiento (cursor del ratón), de valores numéricos o porcentuales.



*Figura 16: Mecanismos gráficos de introducción de valores*

#### 1.4.1.3) Mecanismos de introducción de texto

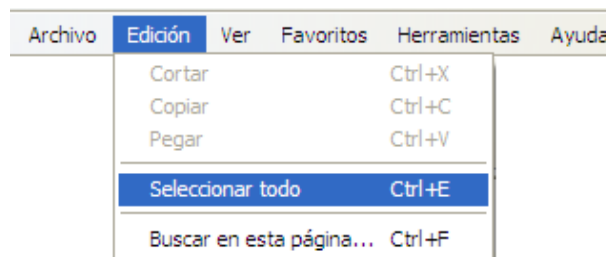
Consiste en la obtención de texto introducido por el usuario. Generalmente se realiza mediante un hardware como el teclado. Existen otras alternativas como son los reconocedores de caracteres (OCR), reconocedores de gestos, de voz, o mediante posicionamiento táctil.

Un aspecto importante relacionado con el texto es su resolución (tamaño en pantalla), que se mide en puntos o píxeles. Se deben utilizar tamaños y tipos de letra que sean legibles y proporcionales a la resolución de pantalla que posea en ese momento el usuario.

#### 1.4.1.4) Mecanismo de selección

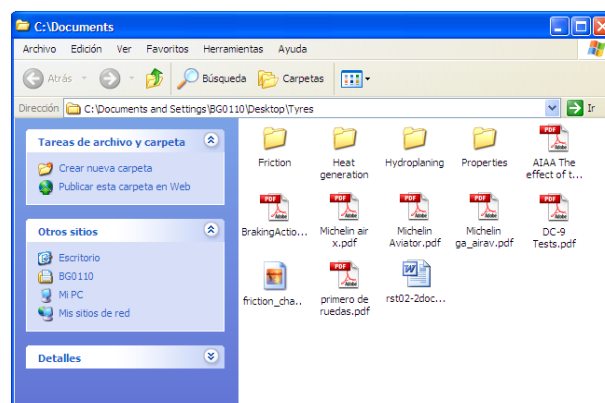
Mediante este mecanismo se obtiene una información que refleja la decisión del usuario. Este elegirá de entre un conjunto de órdenes, atributos, objetos, etc. El conjunto de mecanismos de selección se pueden clasificar en dos tipos según su tamaño:

- De tamaño fijo con elementos invariables (por ejemplo, órdenes). El mecanismo de selección en este tipo de conjuntos utiliza un tipo de representación concreto. El más utilizado es la presentación basada en menús, donde aparecen una serie de ítems que representan órdenes. Se utilizan jerarquías, separaciones, atajos, etc. (Figura 17)



*Figura 17: Menú de selección. (Mecanismo de selección de tamaño fijo y elementos invariables)*

- De tamaño variable, como por ejemplo los objetos que se presentan en la interfaz de la aplicación sometida a estudio. La selección sobre conjuntos de objetos de tamaño variable se organiza sobre listas dinámicas o bien en ventanas (que actúan como contenedores) donde se disponen los elementos con diferentes alternativas de presentación (gráfica, textual, icónica, etc.) y modo de ordenación (Figura 18).



*Figura 18: Ventana contenedora de archivos (Ejemplo de mecanismo de selección de tamaño variable)*

El mecanismo de selección se puede realizar introduciendo un identificador del objeto que se pretenda seleccionar, o mediante el posicionamiento espacial explicado en el apartado anterior, mecanismo de posicionamiento (apartado 1.4.1.1, Cap. 3).

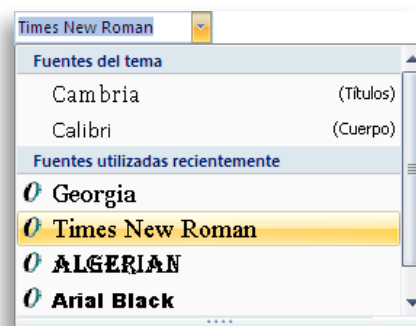
Otras posibles representaciones para los mecanismos de selección pueden ser:

- Botones o conjunto de botones agrupados (barras de herramientas, Figura 19) con representación gráfica, icónica o basada en texto.



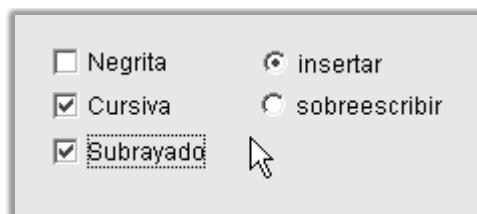
*Figura 19: Barra de herramientas con botones icónicos*

- Listas desplegables o fijas, para seleccionar acciones relacionadas entre sí (Figura 20).



*Figura 20: Lista desplegable, selección de fuente Microsoft Word*

- Elementos de sección lógica, es decir, con dos posibles estados (true/false, seleccionado/no seleccionado, etc.)
- Conjuntos de cuestionarios donde las opciones son mutuamente excluyentes, es decir, solo se puede hacer una selección.



*Figura 21: Selección de opciones de texto*

En cualquiera de los casos el factor más relevante es la ubicación espacial de los elementos de selección. Se ha de buscar que la selección sea rápida, los elementos fácilmente reconocibles y una organización semántica correcta.

#### 1.4.1.5) Mecanismo de arrastre

Este mecanismo consiste en la introducción de una secuencia de posiciones que denotan un movimiento. Generalmente se utiliza para describir explícitamente manipulaciones de objetos gráficos (mover, rotar, escalar).

Ha sido utilizada para dar significado a las operaciones de un sistema de escritorio (“*Drag and Drop*”) y en operaciones de diseño gráfico (Figura 22). Esta tarea requiere de una realimentación continua del objeto desplazado (puede ser una instancia, una línea/caja elástica, un punto de una curva, etc.).

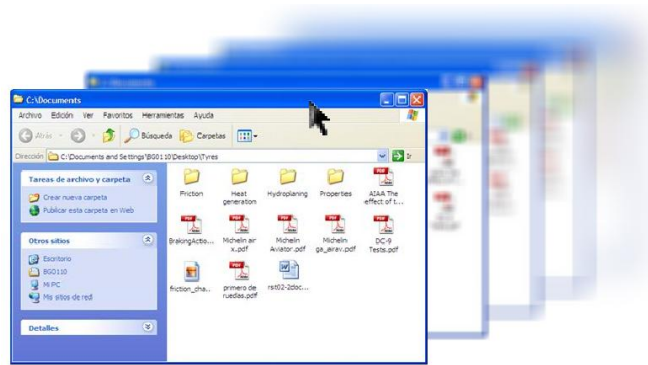


Figura 22: Mecanismo de arrastre de una ventana de Windows

### 1.4.2) Gestión de las entradas del usuario

Una de las funciones más importantes de la aplicación es gestionar la información que previamente ha introducido el usuario. Para ello la comunicación entre usuario y computador es esencial. Los mecanismos que establecen esta comunicación pueden ser: petición de información (*request*), muestreo constante del estado actual (*sample*) o mediante eventos (*events*) [34].

El método de los eventos es el principal mecanismo utilizado por los programas de autor actuales, para el diseño de interfaces interactivas. Cuando el usuario interacciona con los elementos, estas acciones se traducen en eventos que la aplicación interpreta.

Algunos de los eventos más importantes que se pueden encontrar en una aplicación interactiva son:

- Eventos de entrada. Son los generados por el usuario. Algunos de ellos son: eventos del ratón (posición instantánea, selección mediante “click”, desplazamiento mediante el “scroll”, etc.); eventos del teclado (se puede considerar un “array” o matriz de botones de ratón), etc.
- Eventos de las ventanas. Se reciben de la propia ventana donde se reproduce la aplicación. La mayoría de los sistemas de ventanas envían un evento de creación/destrucción, abrir/cerrar, redimensionar, mover, etc.
- Eventos definidos por el usuario. Eventos de alto nivel creados mediante programación software.

### 1.4.2.1) Comunicación entre objetos

Los objetos interactivos son todo aquello que el diseñador pone a disposición del usuario en la aplicación, con el fin de que este introduzca algún tipo de información. Es habitual encontrar objetos relacionados entre sí (como por ejemplo la barra de deslizamiento del texto provoca un movimiento de este). Esta relación o comunicación entre objetos se realiza mediante pseudo-eventos que no son realmente eventos de entrada.

Existen tres modelos básicos de comunicación entre objetos [34]:

- Modelo de llamadas (*callbacks*). Consiste en un procedimiento (algoritmo de programación) que se ejecuta inmediatamente después de un evento. En el lenguaje “*ActionScript 3.0*” por ejemplo, la forma de vincular un procedimiento o función (*function*) a un evento se hace con la siguiente sentencia [15]:

```
Boton1.addEventListener (MouseEvent.CLICK, function);
```

Con esta sentencia se asocia al objeto interactivo “*Boton1*”, cuando sucede el evento “*CLICK*” (pulsar el botón izquierdo del ratón), una secuencia de órdenes con el nombre “*function*”.

- Modelo de notificación. Los componentes notifican a la ventana de nivel superior el evento sucedido. En la ventana raíz se encuentra todo el código que administra los eventos recibidos. Este modelo se utilizaba en las primeras versiones de “*MS Windows*”.
- Modelo de conexión o delegación. Permite la comunicación de un objeto con otro u otros, mediante el registro de objetos receptores de eventos. Este modelo se utiliza en lenguajes como “*Java*” o “*NextStep*”.

### 1.4.3) Diseño gráfico de la aplicación

Este apartado estudia la estrategia para el diseño de la última capa de la aplicación, aquella que el usuario ve y con la cual interactúa (Figura 23). Es el punto de contacto entre el código y el usuario.

La principal cuestión a tener en cuenta para el diseño de IUs, es que el usuario quiere hacer su trabajo de la forma más rápida y sencilla posible, y la herramienta solo debe ser un camino para

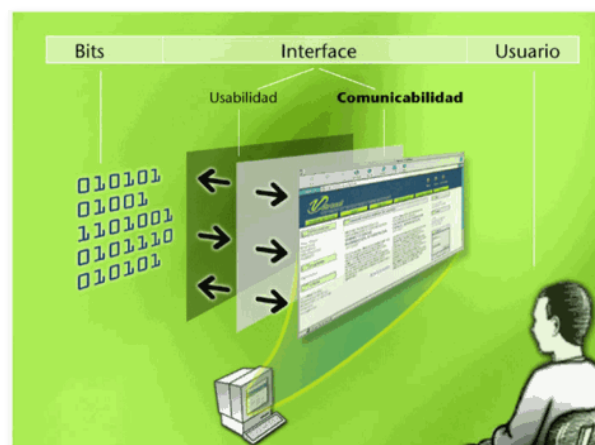


Figura 23: Representación de las partes de una aplicación interactiva



ayudarles a lograrlo. El esfuerzo que se realice en utilizar la herramienta será esfuerzo que se está perdiendo. Esto se resume muy bien con la siguiente cita:

*“Imagina usuarios muy inteligentes pero muy ocupados”; “No importa lo genial que sea tu interfaz, menos es más siempre” (Alan Cooper, “About Face 2.0”) [36].*

Con carácter general, para llegar a diseñar una interfaz usable, es necesario el cumplimiento sistemático de los siguientes tres principios: enfatizar, organizar y hacer reconocible [37].

**Enfatizar**, se refiere al hecho de crear una jerarquía visual en la que los elementos más importantes capten la atención del usuario y generen un flujo de percepción hacia los elementos secundarios. Además, la intención del usuario no tiene por qué coincidir con la que se espera. De esta forma, es importante que el diseñador conduzca la atención del usuario por donde le interesa que vaya. Aquí entra en juego la percepción visual (apartado 2.1, Cap. 3).

**Organizar** el contenido de la interfaz, es decir, establecer relaciones visuales lógicas, fácilmente interpretables por el usuario, siguiendo las leyes y recomendaciones del diseño gráfico de IUs (que se especifican en este mismo apartado).

**Hacer reconocible** una interfaz, es usar elementos como iconos, etiquetas, etc. que requieran menos esfuerzo de comprensión y por lo tanto, facilite la labor del usuario. Es importante tener presente que el usuario debe dedicar toda su concentración a lo realmente importante, el contenido.



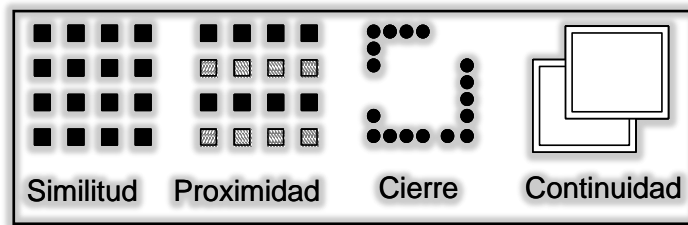
*Figura 24: Principios del diseño visual [37].*

#### 1.4.3.1) Reglas de Gestalt

Uno de los métodos de organización más utilizados por los diseñadores, fue definido por “Gestalt”. Sus reglas se basan en cómo organiza el observador los estímulos visuales, y así establece los siguientes principios [34]:



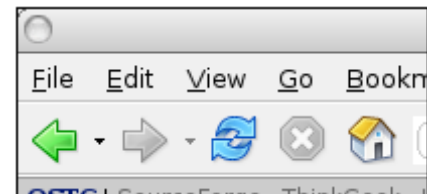
- Similitud. Objetos similares próximos se interpretan como una representación conjunta/agrupada
- Proximidad. Elementos visuales con propiedad común se interpretan como agrupados
- Cierre (clausura). Elementos visuales que tienden a cerrar un área se interpreta como cerrada
- Continuidad (determinación de formas). Discriminación de elementos diferentes según la continuidad natural



*Figura 25: Reglas de Claridad Visual de Gestalt.*

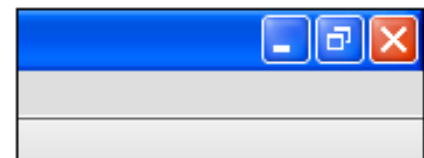
#### 1.4.3.2) Ley de Fitt

Por otro lado, una de las leyes más conocidas en el diseño de interfaces, y a la vez una de las más básicas, es la Ley de “Fitt”. Se basa en la teoría de que cuanto más grande y más cercano al puntero del ratón esté un objeto, más sencillo será pinchar sobre él [36].



*Figura 26: Iconos de navegación grandes*

También es muy importante la posición de los elementos. Se ha demostrado que los objetos que se sitúan en los bordes, son más fáciles de pinchar que los que se sitúan en la zona central (Figura 27). Por ejemplo, un usuario que quiera salir del programa rápidamente, lanza el ratón a la esquina superior derecha, no importando la velocidad ni la precisión del puntero. Sin embargo para pinchar un botón situado en el centro, se requerirá un control de posición y velocidad mayor, y por tanto más tiempo.



*Figura 27: Botones de ventana en el borde de la misma*

### 1.4.3.3) Anticipación

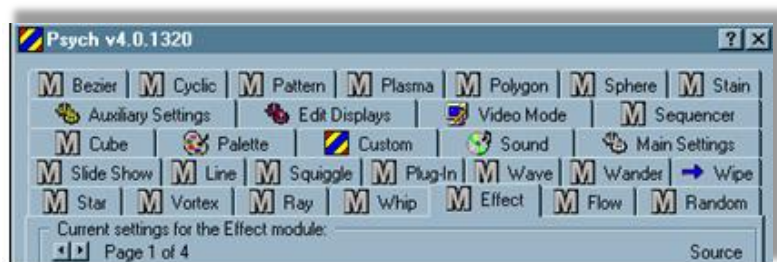
Una herramienta bien diseñada debería intentar anticiparse a las necesidades del usuario y no esperar a que este tenga que buscar la información. En definitiva, reducir la carga de memoria del usuario. Para ello:

- Permitir deshacer, copiar y pegar, mantener los últimos datos introducidos, etc. Aliviar la carga de memoria de corto alcance.
- Elegir de entre un conjunto de listas y no tener que teclear de nuevo para introducir la misma información. Basarse en el reconocimiento antes que en el recuento.
- Proporcionar indicaciones visuales de la posición. Qué está haciendo el usuario, qué puede hacer.
- Proporcionar atajos de teclado.
- Asociar la forma de los objetos a las acciones que realizan.
- Hacer clara la presentación de la información agrupando objetos, evitando demasiada información.

### 1.4.3.4) Autonomía.

La interface y su entorno deben estar a disposición del usuario, deben ser flexibles y muy intuitivas, sin embargo también han de tener ciertas cotas y limitaciones. No se puede dejar nada al azar.

Además es importante que la herramienta incorpore indicadores de estado del sistema que mantengan a los usuarios informados a lo largo de la experiencia.



*Figura 28: Ejemplo de falta de intuitividad.*

### 1.4.3.5) Percepción del color

El color es un elemento muy relevante en el diseño de interfaces ya que ayuda a centrar la percepción visual y mejora la presentación de la interfaz, permitiendo al usuario comprender y manejar la aplicación. Shneiderman (1998) [38] establece catorce pautas claves para la utilización efectiva del color. Las más relevantes son las siguientes:

- Limitar el número de colores utilizados y hacerlo de una forma conservativa. No utilizar más de cuatro o cinco colores diferentes en una ventana y no más de 7 en la interfaz total del sistema.
- Utilizar un cambio de color para mostrar un cambio en el estado del sistema.
- Utilizar el código de colores para apoyar la tarea que los usuarios están tratando de llevar a cabo.
- Utilizar el código de colores en una forma consciente y consistente.
- Utilizar correctamente los pares de colores.
- Si se utilizan muchos colores o si son muy brillantes, el despliegue puede ser confuso.

#### 1.4.3.6) Consistencia

Para lograr una consistencia de la IU es necesario profundizar en ciertos aspectos estructurales de la misma:

- Estructuras invisibles: se requiere que estén definidas claramente y que sigan los estándares del resto de IUs (guías de estilo, apartado 3.3, Cap. 3), si no podría llevar a confusiones al usuario. Un ejemplo son las opciones que utilizan muchas IUs de agrandar las ventanas situando el “mouse” en los bordes. No son normalmente visibles en la interfaz, pero el usuario sabe de su existencia.
- Estructuras visibles: son pequeños símbolos o iconos que puedan ser controlados por el usuario y que faciliten su uso. Por ejemplo, un icono de una impresora para imprimir el contenido.
- Consistencia con la plataforma: la IU debe ser concordante con la plataforma o sistema operativo donde se implementa. Por ejemplo si es una aplicación diseñada para “Windows” debe tener sus estándares, o si es una aplicación web, debe ser posible su navegación, uso de links, etc. Para ello existen las llamadas “guías de estilo”, explicadas detalladamente en el apartado correspondiente (apartado 3.3, Cap. 3).

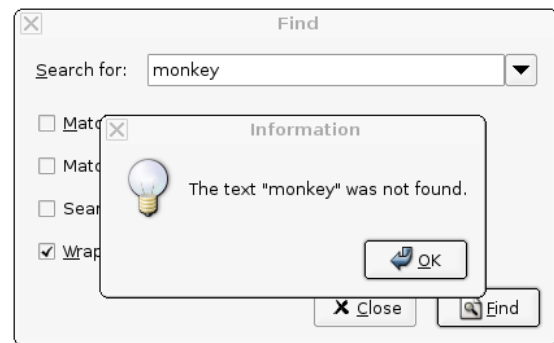
#### 1.4.3.7) Eficiencia del usuario

Es importante considerar la productividad del usuario antes que la productividad de la herramienta. Para ello se debe proporcionar una ayuda clara, es decir, los menús y las etiquetas de los botones deben tener palabras clave del proceso.



*Figura 29: Correspondencia entre iconos y acciones de los botones*

Además, cuando un usuario está trabajando con la herramienta, hay que impedir las interferencias innecesarias, es decir, todo aquello que desvíe su atención e interrumpa el flujo de trabajo. Cada aplicación tiene un elemento clave en el que centrarse, por lo tanto será ese al que más importancia habrá que dar dentro de la IU (Figura 30).



*Figura 30: Interrupción del flujo de trabajo debido a una ventana de información*

### 1.4.3.8) Interfaces explorables

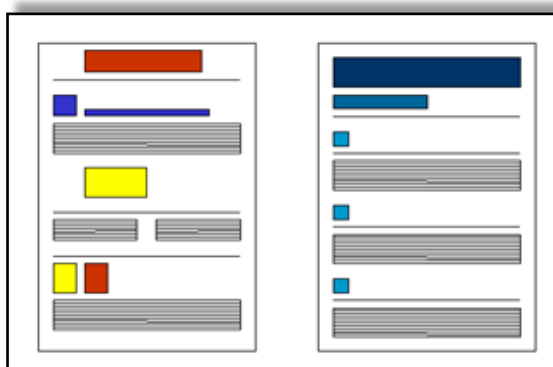
El creador de una IU debe tener en cuenta que al comenzar a utilizar su herramienta, los usuarios no conocerán nada de ella. Por este motivo, es muy importante que estos puedan explorarla sin peligro de llegar a acciones de no retorno o a cerrar la herramienta.

Es conveniente que la interfaz incorpore elementos visuales estables que permitan, no solo un desplazamiento rápido a ciertos puntos del trabajo, sino también un desplazamiento al punto de partida o “home”.

La IU debe poder realizar la inversa de cualquier acción que pueda ser de riesgo. Siempre se debe contar con un comando “deshacer” o retorno.

### 1.4.3.9) Legibilidad

Para que la IU favorezca la usabilidad del sistema, la información que se exhiba en ella debe ser fácil de ubicar y leer. Hay que tener en cuenta factores como la dimensión del texto, la combinación de colores, la presentación visual, etc. Como se verá más adelante (apartado 2.1, Cap. 3), el diseñador debe ser consciente de lo que quiere conseguir del usuario, y ha de guiarle a través de la interfaz por el camino deseado. La única forma de asegurar que el usuario sigue este camino, es mediante una correcta estructuración y legibilidad de la IU.



*Figura 31: Diferencia entre una mala estructuración y uso de colores (Izquierda), y una IU con una estructura legible y clara (Derecha).*

#### 1.4.3.10) Interfaces Visibles

El uso de interfaces invisibles es muy común en las aplicaciones diseñadas para la web. El usuario siempre ve una página específica, pero nunca puede conocer la totalidad del espacio de páginas de internet. La navegación en las aplicaciones debe ser reducida a la mínima expresión. El usuario debe sentir que se mantiene en un único lugar y que lo que varía es su trabajo. Esto evita la necesidad de mantener mapas u otras ayudas de navegación, y además brindan al usuario una sensación de autonomía.

#### 1.4.3.11) Realimentación

La realimentación por parte del sistema interactivo, puede llegar a ser de gran relevancia durante la navegación, ya que el usuario debe estar informado en cada momento de las acciones que realiza. Se trata de mostrar información que le indique qué está haciendo y le ayude a realizarlo correctamente. Se puede utilizar cualquier combinación de canales sensoriales (sonoro, visual, táctil, etc.).

Algunos ejemplos de realimentación son:

- Sobre las órdenes. Mostrar efectos, errores, confirmación, indicadores
- Sobre la selección. Resaltar de forma no ambigua la orden activa

La realimentación debe ser fácil de leer y entender. Para su diseño, se deben estudiar las acciones de cada tarea y ver cómo es la interacción (realimentación del propio gesto), confirmación (selección, mensajes, iluminación) y posibles errores (pantalla de aviso).

Se puede clasificar por su dimensión temporal como:

- Futura. Realimentación de una acción antes de llevarla a cabo. Indica qué sucederá si se realiza una acción (por ejemplo, etiqueta informativa de los botones).
- Presente. Realimentación durante la interacción. Indica lo que está sucediendo actualmente (p. e. borrado de ficheros, formateando, mover cursor).
- Pasada. Información de lo que ha sucedido, y cómo ha cambiado el sistema (p. e. información de finalización de tareas).

### 1.4.3.12) Gestión de errores

Un factor crítico en el flujo de los sistemas interactivos (desde el punto de vista del usuario), es cómo se organizan los mensajes de error y su explicación. Normalmente ocurren por un desconocimiento motivado por las siguientes causas:

- Errores por acciones del usuario. Confusión entre la intención del usuario y su acción (intención correcta, pero realización incorrecta). La solución es mejorar el diseño ergonómico, mejorar los aspectos físicos del diseño (ubicación de menús, tamaño, posición, visibilidad, color, etc.)
- Errores por las intenciones del usuario. El usuario realiza una acción equivocada. El programador no se ha anticipado correctamente a la intención del usuario. La solución es mejorar el modelo mental del programador (apartado 2.3, Cap. 3).

El usuario, ante un error, debe reconocerlo, para evitar confusión. Para ello, es necesario que el diseñador informe correctamente al usuario de la acción incorrecta. Algunas técnicas que se deben evitar en los mensajes de error son:

- Tono imperativo. Aumenta la ansiedad del usuario. Dificulta la corrección del error ("acción ilegal", "error fatal").
- Mensajes genéricos o confusos. Ofrecen poca información ("error sintáctico", "error").

Una aplicación convenientemente diseñada, requiere de un estudio para hallar la distribución y frecuencia de errores y problemas. Este estudio, también llamado evaluación, se ha de realizar durante todo el proceso de creación de la aplicación (información más detallada en el apartado 1.6, Cap. 3) Es conveniente además, seguir la normativa y los estándares establecidos para el diseño de interfaces amigables (apartado 3, Cap. 3) y proveer al usuario de documentación referente al uso de la aplicación (Anexo I).

## 1.5) Prototipos

La creación de un prototipo es una fase clave en el proceso de desarrollo de una interfaz de usuario. Como se explica en el apartado de ciclo de vida de una aplicación interactiva (apartado 1.1, Cap. 3) el prototipado transcurre paralelamente al proceso principal y afecta a cada una de las etapas de este.

En este apartado se definirán los tipos de prototipos que se han de crear para sacar el máximo rendimiento a este proceso, así como las diferentes técnicas de prototipado con sus ventajas y sus inconvenientes.

En la fase de creación de un prototipo, tanto el diseñador como la aplicación juegan papeles claves. El diseñador debe tener claro qué parte de su aplicación quiere evaluar, y conforme a esto, diseñar el prototipo que más se adecúe a sus intereses. Así surgen dos tipos de prototipos [35]:

- Prototipado vertical: incorpora toda la funcionalidad de la aplicación dejando de lado el diseño. Con este prototipo, el diseñador quiere evaluar el funcionamiento, la navegación y la interactividad de la interfaz, sin importarle su apariencia. En definitiva se quiere verificar la correcta programación de la aplicación.
- Prototipado horizontal: al contrario que el vertical, en este prototipo se tiene muy en cuenta la apariencia y la presentación de la interfaz. Se busca la máxima semejanza al producto final, sin embargo no incluye funcionalidad alguna.

Por otro lado, con interfaces de usuario complejas, compuestas de muchas partes, es habitual que el diseñador no pueda realizar un prototipo fiel al producto final y que abarque toda la funcionalidad que debería tener. No es una hazaña imposible hacer un prototipo completo, sin embargo no sería efectivo desde el punto de vista económico y temporal. Es necesario que exista un balance entre la utilidad del prototipo y el desembolso económico que supone su creación. De esta manera el diseñador ha de elegir entre dos clases de prototipos a realizar:

- Prototipos de baja fidelidad: se trata de un prototipo basado en los aspectos generales de la IU. No entra en detalles, de tal forma que puede abarcar un gran espectro. Posee diversas ventajas: su creación es rápida y fácil de cambiar; permite la evaluación de los elementos del diseño como iconos, colores, apariencia, etc.; es útil para identificar requisitos; genera unos costes bajos.

En el lado opuesto, encontramos una serie de desventajas: Es un prototipo con una navegación e interacción limitada; Con requisitos bien establecidos (en la primera etapa de creación de la IU) se vuelve un prototipo inútil; No permite la evaluación de la programación.

- Prototipos de alta fidelidad: Se trata de un prototipo basado en una parte concreta de la interfaz. Abarca mucho menos espectro, pero detalla con la mayor precisión posible esa parte. Las principales ventajas de este prototipo son: incorpora una funcionalidad e interactividad completa; su aspecto es muy parecido al sistema final; sirve para evaluar al detalle cada uno de los componentes de la interfaz.

Por el contrario, las desventajas que presenta son las siguientes: el coste es más elevado; es más difícil de crear y de cambiar, por lo que lleva a una mayor pérdida de tiempo; no es efectivo si no se tienen previamente los requisitos bien establecidos.



A continuación se presentan algunas técnicas para la creación de prototipos basados en interfaces de usuario. Todas ellas presentan una serie de ventajas e inconvenientes. Algunas dan más importancia a la presentación y otras a la interacción (“look & feel”). El diseñador ha de elegir la que se adapte mejor a sus objetivos [34].

- **Bocetos:** Se trata de prototipos estáticos. Son las primeras ideas sobre lo que se pretende diseñar. Se usa principalmente en la etapa inicial del diseño cuando se busca establecer la funcionalidad de la interfaz, pasando por alto la estética. Son prototipos rápidos y baratos, sin embargo carecen de detalles.

- **Storyboards navegables:** Se trata de un prototipo estático parecido a un boceto pero que abarca un espectro mayor. Viene de la industria cinematográfica y se basa en una serie de bocetos que simulan el flujo real de la interfaz o una parte de ella. Generalmente se usa en las etapas iniciales. No busca ser fiel a la estética sino que muestra de una forma muy esquemática y clara la navegación de la interfaz y todas las posibilidades de interacción. Gracias a este prototipo el diseñador es capaz de ver todos los posibles caminos que puede tomar el usuario cuando use la aplicación. Además resulta barata y rápida de crear y modificar. Sin embargo, no entra en detalles técnicos (Figura 32).

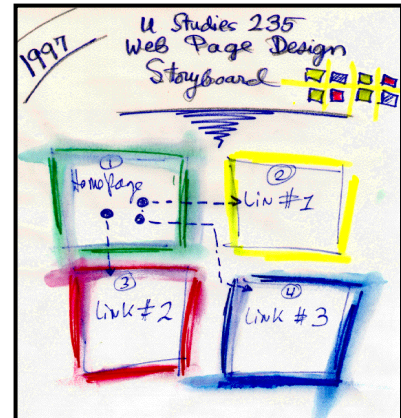


Figura 32: Ejemplo de prototipo StoryBoard para una página Web

- **Maquetas digitales:** Se trata de prototipos estáticos o dinámicos, muy fieles con la parte que pretenden evaluar. Intentan imitar la estética final de la interfaz, de tal forma que se consiga un alto detalle del diseño gráfico. Requiere más tiempo para su creación, más conocimiento por parte del diseñador y más coste económico. Es efectiva en las etapas de diseño
- **Prototipos de software:** Se trata de crear una versión preliminar de la interfaz a la que se quiere llegar. Son prototipos dinámicos y robustos, con una alta fidelidad. Se necesitan conocimientos del sistema y el empleo de más tiempo que con otros prototipos. Sin embargo permite evaluar el rendimiento de la interfaz final, su navegación y su interacción (no tanto su estética y diseño gráfico).



Figura 33: Ejemplo de prototipos software



## 1.6) Evaluación

Como se definió en el apartado de ciclo de vida de una aplicación (apartado 1.1, Cap. 3), el proceso de evaluación de una interfaz gráfica se debe realizar a lo largo de su ciclo de vida. A diferencia de lo que parece normal pensar, no es un proceso que se realice justo antes de la fase de lanzamiento, sino que se debe evaluar el funcionamiento y la presentación constantemente. De esta forma, el control antes del lanzamiento de la interfaz, se realiza sobre los prototipos.

En el proceso de evaluación, tanto la aplicación como el usuario adquieren gran relevancia. El objetivo fundamental es identificar las necesidades de los usuarios y conocer su grado de satisfacción con la aplicación. Esta información se consigue mediante la realimentación del usuario. Se han de realizar estudios de aceptación y encuestas posteriores al lanzamiento de la interfaz.

Por otro lado, es necesario evaluar los errores de los prototipos de tal forma que se puedan aplicar las mejoras y soluciones al producto final. Algunas de las pautas para evaluar las interfaces de usuario son las siguientes [37]:

- Visibilidad del estado del sistema.
- Semejanza del sistema al mundo real.
- Control y libertad por parte del usuario.
- Consistencia y estandarización.
- Prevención de errores.
- Reconocimiento de acciones y opciones.
- Flexibilidad y eficiencia en el uso.
- Estética y diseño minimalista.
- Reconocimiento de errores, diagnóstico y recuperación.
- Ayuda y documentación.

Estos son indicadores que permiten al diseñador realizar una medición de la calidad de su interfaz. Sin embargo, no son los únicos. No existe un acuerdo total en la cantidad de indicadores que debe usar un diseñador para evaluar su interfaz o de cuáles son los más relevantes.

Para solucionar este problema, se crearon una serie de estándares y guías de estilo que ofrecen al diseñador recomendaciones, principios y directrices para elaborar una interfaz de

calidad. Todos estos conceptos se explican con claridad en el apartado de normativa y estándares (apartado 3, Cap. 3. Es importante mencionar la existencia de diversas normas relacionadas con la evaluación y control de las interfaces:

- ISO/IEC 9126: “Evaluación de productos software – características de calidad y directrices para su uso”. Sus cuatro partes establecen métricas para evaluar un producto software.
- ISO 9241: “Requisitos ergonómicos para trabajar con terminales de presentación visual (*Visual Display Terminals*, VDTs)”. En especial la parte 11 proporciona una guía para la realización de los ensayos de resultado obtenidos por el usuario, con el objeto de evaluar los sistemas.
- ISO 11064: “Diseño ergonómico de centros de control”. En especial la parte 7: Principios para la evaluación de los centros de control.
- ISO/IEC 14598: “Evaluación de productos software”.

Hasta ahora se han definido los indicadores para evaluar la interfaz, pero no se ha hablado de cómo evaluar un error detectado. Estos surgen tras la creación de los prototipos y han de ser clasificados. Cuanto más compleja sea una interfaz, más riesgo habrá de obtener errores y problemas. Ante diversos errores es necesario establecer su grado de severidad e incidencia en el resultado final con el fin de corregirlos. Los factores que determinan el grado de severidad de un problema son [35]:

- La frecuencia de ocurrencia.
- El impacto que causa la ocurrencia del problema.
- La persistencia del problema.
- El impacto en el mercado.

Después de realizar este análisis, es necesario evaluar el problema. Para ello se propone un sistema de medición de errores consistente en una escala del 0 al 4 en orden creciente de importancia.

- 0 – No debe considerarse problema o se debe dar una prioridad de solución baja.
- 1 – Problema estético con una prioridad baja. Se solucionará si se dispone de tiempo extra para realizar el proyecto.
- 2 - Es un problema menor. Su corrección tiene prioridad media y ha de resolverse para conseguir una eficacia óptima de la interfaz.

- 3 - Es un problema mayor. Su corrección tiene prioridad alta y ha de resolverse para que la interfaz funcione correctamente
- 4 - Es una catástrofe. Su prioridad es muy alta, debe ser resuelto inmediatamente para poder continuar con el proceso de diseño.

### 1.5) Conclusiones

A la hora de enfrentarse al diseño de una aplicación o sistema interactivo, se presentan gran cantidad de dificultades no solo por la complejidad en la programación, sino también en la presentación a los usuarios. Esta última fase es de gran relevancia y a menudo los diseñadores desconocen las técnicas para la creación de una interfaz de usuario efectiva.

En este apartado se abordan las fases que ha de seguir un diseñador para la creación de una interfaz de usuario. Es un proceso cíclico y en constante movimiento, por lo que el diseñador, aplicación y usuario volverán a pasar por las mismas fases mientras dure la vida útil de la aplicación

Profundizando en cada una de las fases se definen todos los tipos de mecanismos de que dispone el diseñador para presentárselos al usuario y recibir información de él. Este proceso es vital ya que será la única forma que tiene el diseñador de recibir la información y anticiparse a ella correctamente. De una forma paralela al proceso de diseño, se han de crear prototipos y evaluarlos con el fin de obtener el producto final de la forma más rápida, menos costosa y habiendo cumplido todos los requisitos propuestos.

Pese a que existe una amplia bibliografía en cuanto al diseño de interfaces de usuario, en su gran mayoría son una recopilación de recomendaciones basadas en la experiencia de autores experimentados, por lo que el diseño de una aplicación siempre es una labor muy compleja.

## II) EL USUARIO Y EL ENTORNO DE LA APLICACIÓN

### 2.1) Percepción visual

En el uso de aplicaciones interactivas, el canal visual es la forma más general de percepción de información que posee el usuario. Por este motivo, el diseño gráfico de la interfaz es el que condiciona la fluidez de transmisión de información entre aplicación-usuario.

El diseño gráfico debe estar basado en la forma en la que el usuario percibe visualmente la información. El ser humano realiza la percepción visual de una forma parecida a una cámara, sacando instantáneas cada intervalo de tiempo de entre 30 a 120 ms. El ojo se mueve dando “saltos” llamados “sacadas” y tiene momentos de fijación en un punto concreto donde se focaliza la atención (Jacob, año 1995) [37]. El resto de información dentro del campo visual se percibirá con una atención inversamente proporcional a la distancia que la separa del punto focalizado. Es la llamada visión periférica (Ware, año 2008) [37].

La primera percepción que recibe el usuario es masiva: color, movimiento, tamaño, organización, etc. (Wolfe, año 2006) [37]. A pesar de que el usuario dirige su atención hacia “lo que quiere ver” le resulta inevitable percibir cierto tipo de información visualmente atrayente, aún estando en el campo de visión periférica. Esta explicación llevada al campo de las interfaces de usuario, tiene gran importancia. El usuario prestará atención a aquellas cosas sobre las que realiza la búsqueda. Sin embargo el diseñador es capaz de captar su atención estableciendo “puntos de entrada” (Fidalgo, año 2006) [37].

Cuantas más diferencias gráficas tenga un elemento con respecto a los colindantes, mayor será la captación del usuario (tamaño, color, movimiento diferente al resto).



*Figura 34: La atención se ve guiada al círculo rojo porque es el único inusual.*

*Fuente: [www.human-computer.net/blog](http://www.human-computer.net/blog)*

Un ejemplo muy claro de esta teoría son los llamados “banners” o publicidad que incorporan numerosas interfaces web. Su objetivo es captar la atención de los usuarios para transmitirles información publicitaria.

Posterior a la atención visual, se produce una organización perceptual de lo que se está viendo. El usuario, automáticamente, organiza las formas, los objetos, la información y en general la sintaxis gráfica de lo que percibe. En esta fase entran en juego las comentadas leyes de Gestalt (apartado 1.4.3.1, Cap. 3). Estas leyes aplicadas correctamente, permitirán al usuario realizar un mayor número de operaciones automáticas, traducándose en una mayor facilidad de uso.

La última fase de la percepción visual corresponde con la vinculación al conocimiento. Consiste en que el usuario intenta reconocer o dar significado a aquella información que haya percibido. En esta fase entra en juego el conocimiento de usuario y la teoría de cognición humana (lenguaje, memoria, razonamiento, etc.) explicada a continuación.

## 2.2) Cognición humana

*“Se define como cognición humana, la facultad de procesar información a partir de la percepción, el conocimiento adquirido y características subjetivas que permiten valorar la información.” [39].*

El estudio de este proyecto busca como objetivo principal, transmitir una información a un usuario a través de una herramienta o aplicación interactiva. El problema recae en la incertidumbre a la hora de diseñar esta aplicación, debido al alto grado de complejidad que tienen los seres humanos para recibir información. La única forma de comprender este proceso es mediante el conocimiento empírico de la cognición humana, estudiado por la psicología. Cuanto más se conozca de él, mejores serán las decisiones que tomará el diseñador a la hora de crear su aplicación.

## Almacenamiento de la información.

Antes de estudiar el proceso cognitivo, es necesario conocer los fundamentos de la percepción de la información. La memoria del ser humano se divide en dos partes: una memoria operativa a corto plazo y otra a largo plazo [37]. La primera guarda relación directamente con la consciencia (razonamiento y comprensión). La segunda es una memoria estable de almacenamiento de límites desconocidos.

Debido a las características de la memoria operacional, el ser humano está limitado a la hora de percibir elementos. Así en una interfaz gráfica, un usuario no será capaz de percibir más de 3 o 5 elemento. A esta limitación se le une el esfuerzo que ha de emplear el ser humano para almacenar información en la memoria a largo plazo. Este esfuerzo se genera al sintetizar la información, ya que almacenar datos inconexos, es mucho más complicado. Este es un factor clave a la hora de tomar decisiones en el diseño de una interfaz.

La navegación que hace el usuario por la interfaz es de “ensayo y error”. Debido a la dificultad e incapacidad de asimilar todos los elementos de la interfaz, el usuario prestará su atención al primero que crea que se corresponde con su búsqueda (aún siendo el incorrecto). Esto se traduce en pérdida de tiempo y de esfuerzo. Por este motivo, es importante que el diseñador enfatice gráficamente y ordene de forma correcta los conjuntos de elementos de la interfaz.

## Toma de decisiones

La toma de decisiones es una acción cognitiva compleja, influida por multitud de factores, que se divide en dos sistemas: intuitivo y racional. El primero es rápido, susceptible a errores, basado en la experiencia y en la emoción. En cierto modo se trata de un sistema de toma de decisiones que no se produce bajo la consciencia. El segundo es un sistema lógico, lineal y consciente, sin embargo emplea más tiempo y esfuerzo.

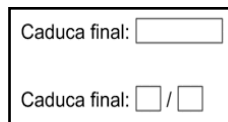
El uso de uno u otro, depende de cómo el usuario desee “economizar” su esfuerzo. Por este motivo, gran parte de las decisiones que tome el usuario dentro de la interfaz, las hará de forma intuitiva. Este factor es importante para determinar que, en el proceso de diseño, no se debe dejar al usuario que tome decisiones importantes, porque no lo hará. Un usuario en contacto con una interfaz, no busca comprender el diseño de esta, sino que emplea su esfuerzo cognitivo en satisfacer sus objetivos.

## Prevención de los errores

La filosofía del diseñador, no debe ser impedir que los usuarios cometan errores, porque según se ha definido anteriormente, el sistema intuitivo de toma de decisiones es propenso a desembocar en un error. El diseñador debe intentar prevenirlos y buscar soluciones cuando ocurran.

Para ello, algunos de los principales recursos son:

- Limitar las acciones del usuario y la toma de decisiones. El usuario hará intuitivamente lo que el diseñador quiere que haga (Figura 35).



*Figura 35: Mayor limitación en la opción de abajo.*

- Ofrecer al usuario una orientación o sugerencia de cómo debería actuar para minimizar el esfuerzo en caso de tener que tomar una decisión “racional” (no intuitiva) (Figura 36).




*Figura 36: Sugerencias en la búsqueda.*

- Solicitando confirmaciones después de tomar una decisión que pueda perjudicar el proceso o que haya podido ser tomada de forma intuitiva e incorrectamente.



*Figura 37: Mensaje de confirmación de GMai.*

- Permitir deshacer o rehacer la operación llevada a cabo. De esta forma se concede al usuario otra oportunidad en caso de haber tomado la decisión incorrecta.
- Ofrecer soluciones automáticas a los errores



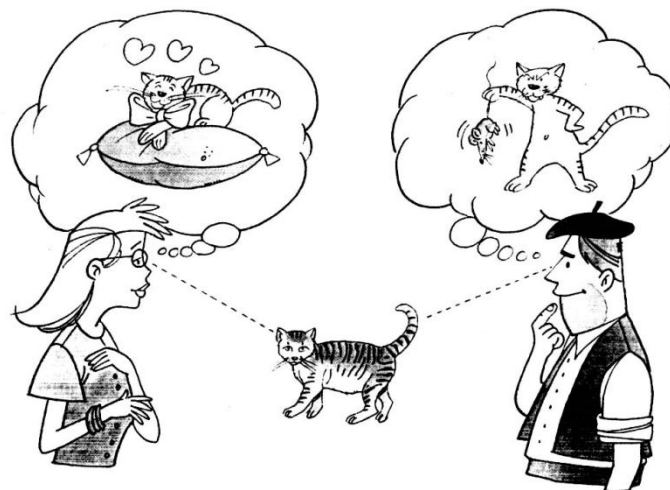
*Figura 38: Corrección en la búsqueda de Google*

Existe muchos otros recursos como: advertir al usuario antes de tomar una decisión importante, evitar la pérdida de información de forma accidental, proporcionar mensajes donde se detalle el error cometido de una forma clara, etc. Lo primordial es conseguir que el usuario no se colapse ni se sobre esfuerce, haciendo la navegación por la interfaz amena e intuitiva.

## 2.3) Modelos mentales

La principal cuestión que determina el diseño de interfaces es, cómo piensan los usuarios (Figura 39). Su forma de actuar, guiada por su conocimiento y su experiencia, está basada en su modelo mental. Un usuario que no entiende la información de la interfaz, no hará un uso correcto de ella. De esta forma el diseñador debe crear una interfaz con un modelo conceptual que permita al usuario reconocer y saber cómo utilizar la información.

Los conocimientos que posee el usuario provienen de su experiencia en la utilización de otras interfaces. Esta norma, pese a estar generalizada, no se puede tener rigurosamente en cuenta. El conocimiento y los modelos mentales de cada usuario pueden ser muy diferentes y están en continuo cambio.



*Figura 39: Distintos modelos mentales sobre un mismo objeto*



Para que el usuario construya un modelo mental de la interfaz correcto y acorde con el del diseñador, este último debe saber construir su propio modelo mental. Así se conseguirá una interfaz entendible y usable. La labor, por tanto del diseñador, será crear un modelo de diseño eficiente y bien definido para el tipo de usuario al que se dirige. Esto se consigue mediante el empleo de mapas conceptuales.

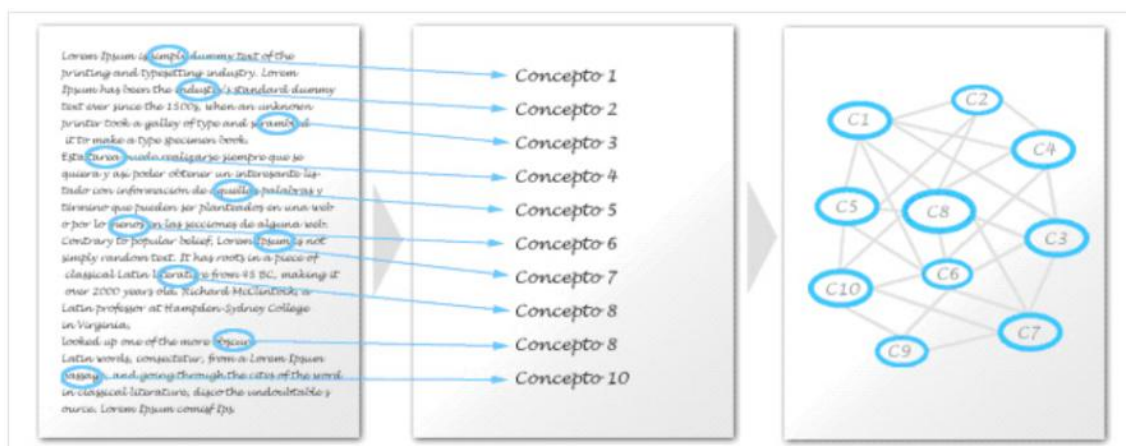


Figura 40: Extracción de un mapa conceptual a partir de un texto

## 2.4) El entorno del usuario

Cuando el usuario se enfrenta a una interfaz, solo conocerá los elementos que la componen si ya los ha visto antes y conoce su significado. El diseñador construye su interfaz a partir de convenciones o estándares ya creados, que aportan significados compartidos y contruidos colectivamente. Así por ejemplo, un usuario reconocerá rápidamente el significado de las dos primeras imágenes de la siguiente figura (Figura 41), pero empleará más tiempo en deducir la tercera, simplemente por el hecho de que no está acostumbrado a ella.

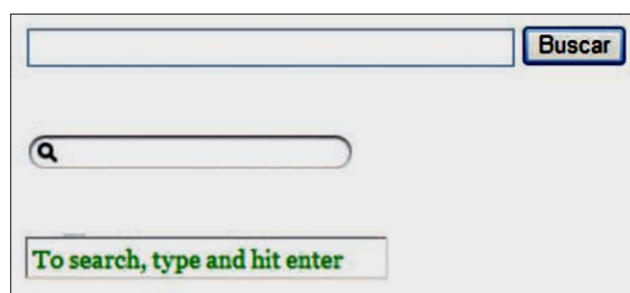


Figura 41: Ejemplo de convenciones para la búsqueda (las dos primeras)

Estos conocimientos no son universales sino que dependen del entorno de usuario y del contexto donde se desarrolle la interfaz. El diseñador y el usuario han de negociar un mismo significado para los elementos de la interfaz. Así la efectividad de los elementos no se mide por su detalle gráfico o su grado de interactividad, sino en cuánto se asemeja a lo que un usuario está acostumbrado a ver. No es viable diseñar una interfaz con las experiencias propias o la perspectiva personal del diseñador.

Aunque la visión de cada uno de los usuarios sea subjetiva, se ha llegado a un conjunto de acuerdos comunes y convenciones compartidas socialmente. Intentar conseguir una interfaz gráficamente muy elaborada, puede producir una falta de reconocimiento por parte del usuario. Esto se denomina “exceso de originalidad” (Costa, año 1994) [37]. El usuario ha de sentirse en un entorno cómodo, donde encuentre elementos habituales.

Los tipos de elementos que pueden resultar a un usuario como “habituales” o “familiares” dependen de la situación geográfica, del conocimiento previo del usuario, del contexto sociocultural, etc. Sin embargo, el diseñador puede encontrar recomendaciones generalizadas en los estándares y normas de diseño de interfaces (apartado 3, Cap. 3).

## 2.5) DCU (Diseño Centrado en el Usuario)

El Diseño Centrado en el Usuario es la optimización y adaptación del producto al usuario. Fue definido por la UPA (*Usability Professionals Association*) y afirmaba que para conseguir un producto adaptado al usuario era necesario llevar a cabo investigaciones en antropometría, ergonomía, arquitectura o biomecánica. El objetivo es conseguir que el diseño no esté reñido con el placer de uso [40].

El término de DCU no significa usabilidad. La diferencia es, que mientras que la usabilidad es un atributo de la calidad del diseño, el DCU es un camino para llegar a un producto usable. Actualmente, con el enorme desarrollo de la tecnología, el diseñador no ha de buscar explotarla al máximo, sino conducirla y adaptarla a las necesidades del usuario mediante el DCU.

Los elementos que aparecen en el DCU para sistemas de información digital, fueron esquematizados por Garret en el año 2002 [40] y se muestran en el siguiente esquema (Figura 42). Los planos que se muestran, se deben seguir de abajo hacia arriba. El primer plano incorpora lo que los usuarios esperan de la interfaz, los objetivos y en definitiva la estrategia a seguir. El siguiente plano define la estructura de la interfaz y el esqueleto como plasmación de la estructura abstracta. A continuación, se muestran los planos con los elementos de interacción, navegación y por último, la capa de la superficie que es el reflejo de todas las capas anteriores.

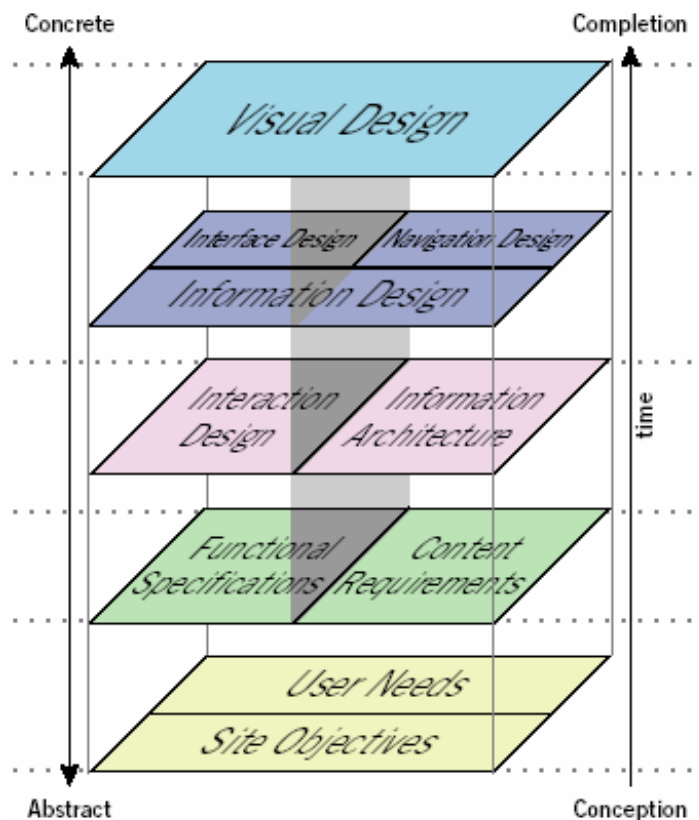


Figura 42: Planos y elementos en el diseño centrado en el usuario (según Garret, 2002) [40].

El proceso para un diseño que ponga al usuario en el centro del sistema, viene definido en la norma ISO 13407, y se explica en el apartado de estrategia de diseño (apartado 1.4, Cap. 3). Sin embargo el DCU no debe ser entendido como una consecución de pasos que hay que seguir estrictamente, sino que se trata de una filosofía de diseño. El DCU implica entender cómo el individuo usará el producto y cómo la aplicación resolverá las necesidades reales del usuario.

Las acciones del ser humano vienen motivadas por sus necesidades. Una interfaz que no satisfaga al menos una necesidad real, no tendrá sentido. La forma de encontrar esas necesidades, es mediante la información y la investigación (Figura 43).



Figura 43: Esquema de la labor del diseñador y del usuario.

Existe una metodología para llevar a cabo el diseño centrado en el usuario. Se trata de diversas técnicas con un objetivo común: conocer y comprender las necesidades, limitaciones y comportamientos del usuario.

Algunas de estas técnicas son las siguientes:

- **Test a los usuarios.** Es una de las formas más efectivas de evaluar el comportamiento de un usuario ante la interfaz o prototipo y encontrar sus necesidades. Esta técnica es muy productiva, ya que el diseñador pierde la perspectiva de su trabajo al estar expuesto constantemente a él y no es capaz de valorarla objetivamente. Presenta algunos problemas como alto coste de reclutamiento de participantes, de tiempo y de esfuerzo dedicado. Otro problema es, que se fuerza al participante a un contexto y a unas necesidades que no tienen por qué ser compartidas. Para evaluar la aplicación diseñada en este proyecto, se realizará esta técnica mediante la creación de un formulario para los ejercicios (apartado 4, Cap. 4)
- **Evaluación heurística:** Es una técnica propuesta originalmente por Olich y Nielsen en el año 1990 [37]. Es similar a los test, con la diferencia de que los llevan a cabo participantes expertos. Estos, aplican los principios del diseño usable y los requisitos que establecen los estándares. Con esta técnica se detectan una mayor cantidad de problemas de usabilidad que son de menor importancia (más específicos). En ocasiones, puede no resultar eficaz.
- **Eye-tracking:** Se trata de una técnica que estudia la percepción visual del usuario. Automáticamente, al explorar una interfaz, el usuario atiende a zonas específicas de la misma. Este comportamiento, a menudo, se ha interiorizado a partir de su experiencia navegando por otras interfaces. Además se aplican las reglas de la percepción visual (apartado 2.1, Cap. 3). Actualmente los sistemas de *Eye-tracking* son muy sofisticados pero de alto coste económico. Por otro lado su interpretación no es tan sencilla. No basta con analizar que regiones de la interfaz tienen mayor densidad de percepción (Figura 44), sino que es importante analizar las cualidades del individuo, su conocimiento, etc. Es una técnica como los test de usuarios pero con un grado de sofisticación más elevado (algunas veces innecesario).



Figura 44: Mapas de calor sobre tres interfaces diferentes. Fuente: Nielsen (año 2006) [37].

- Etnografía: Técnica por la que se estudia y describe científicamente la conducta, el comportamiento, las creencias y las acciones de un usuario o de una sociedad. Es importante convivir en el ámbito sociocultural y en el contexto donde está inmerso el usuario para comprender su comportamiento. Esta técnica no es tan eficaz en la evaluación del producto final, como en el proceso de análisis de requerimientos al comienzo del proceso de creación.

Otras técnicas también empleadas para llegar a conseguir un diseño centrado en el usuario son: entrevistas, encuestas, estudios de mercado, etc.

## 2.6) Conclusión

En este apartado se ha explicado el sentido último de la creación de interfaces. Puesto que toda aplicación funciona como un medio de transmisión información, es importante conocer cómo el ser humano la recibe y la interpreta.

Cada ser humano actúa de forma diferente, no solo por las diferencias biológicas, sino también por el contexto socio-cultural, la experiencia, las necesidades, etc. Pese a esto, existe un patrón de comportamiento que engloba a la gran mayoría (siempre será imposible englobar a todos). La labor del diseñador será encontrar este patrón. Para ello dispone de técnicas de diseño centrado en el usuario y de normas y estándares que se definirán en el siguiente apartado (apartado 3, Cap. 3).

Conociendo cómo funciona el ser humano, el diseñador podrá crear una interfaz que cumpla sus objetivos, que sea eficaz y que además satisfaga las necesidades de quien la usa.

## III) NORMATIVA Y ESTÁNDARES

### 3.1) Introducción

Cuando empezaron a desarrollarse las técnicas de creación de software, el diseño de aplicaciones se vio muy influenciado por las limitaciones de los ordenadores. Se tenía poca consideración hacia el usuario, el cual tenía la obligación de adaptarse a la aplicación. Hoy en día se ha cambiado este concepto, siendo necesario adaptar la aplicación al individuo, guiándole en el proceso de aprendizaje y haciendo que el proceso sea lo más sencillo y ameno posible.

Para llegar a este nivel es necesario conocer el modelo mental del individuo, sus habilidades psíquicas, psicológicas y físicas, (apartado 2, Cap. 3) tarea muy complicada para el diseñador del software. Por este motivo, los expertos en la materia, han consensuado una serie de normas y principios generales sobre el diseño de interfaces de usuario, con el fin de que los diseñadores puedan conseguir la máxima eficacia en sus aplicaciones.

Así surgieron las reglas de diseño formadas por estándares y directrices. Los estándares son reglas estrictas de uso específico, mientras que las directrices son convenciones sugeridas. Un ejemplo de directrices son las guías de estilo, publicadas por los sistemas de interfaces gráficas de usuario, que dan sugerencias para programar en entornos específicos.

### 3.2) Estándares

*“Un estándar es un requisito, regla o recomendación basada en principios probados y en la práctica. Representa un acuerdo de un grupo de profesionales oficialmente autorizados a nivel local, nacional o internacional.” [41]*

Los estándares podrán ser por lo tanto de carácter: local (consenso aceptado desde una industria, organización profesional o entidad empresarial); nacional (consenso aceptado por varias organizaciones de una misma nación); internacional (a nivel mundial).



En el campo de la informática, existen estándares tanto para aplicaciones software como para elementos hardware. Así, se puede encontrar estandarizado el tamaño y forma de la pantalla, los cables de conexión, la inclinación del teclado, etc. En cuanto al software, la mayoría de los estándares están relacionados con las interfaces de usuario. Estos buscan asegurar que los factores humanos de calidad estarán incorporados en el sistema.

Un estándar software establece [42]:

- Una terminología común.
- Fácil mantenimiento y actualización, ya que todos los sistemas tendrán la misma estructura y estilo.
- Una identidad común que facilita su reconocimiento y reduce la necesidad de formación del usuario.
- Mayor seguridad y menor riesgo de comportamiento inesperado al haber pasado controles de calidad.

Existen dos tipos de estándares: los estándares “de facto” y los estándares “de iure”.

Los estándares “de facto” se crean a partir de productos de la industria que tienen gran éxito en el mercado, o a partir de desarrollos de investigación universitaria con gran difusión (por ejemplo el sistema operativo *Windows*)

Los estándares “de iure” están creados por un comité con estatus legal y además están avalados por un gobierno o institución para producir estándares.

En informática existen gran cantidad de comités que han creado estándares, entre los que se encuentran: ISO (*International Organization for Standardization*), IEC (*International Electrotechnical Commission*), ANSI (*American National Standards Institute*), IEEE (*Institute of Electrical and Electronics Engineers*), CEN (*Comité Européen de Normalisation*), W3C (*World Wide Web Consortium*), etc.



Figura 45: Logotipos de los comités y organizaciones más importantes de creación de estándares y normas

### 3.2.1) Estándares de iure para la IPO (Interacción Persona-Ordenador)

La publicación de estándares relacionados con la Interacción Persona-Ordenador es relativamente reciente y se ha llevado a cabo en su mayoría por la organización ISO.

Actualmente hay gran cantidad de estándares publicados por esta organización y otros muchos en proceso. Su elaboración y validación es un proceso complejo y largo en el que los documentos van pasando por diferentes estados. Para generar un estándar ISO habrá que pasar por el procedimiento que se muestra en la Tabla 5.

*Tabla 5: Estados implicados en el proceso de desarrollo de un estándar ISO*

Estado	Tipo de Documento	Documento	Descripción
1	AWI	Approved Work Item (Elemento de trabajo aprobado)	Previo a un borrador de trabajo
2	WD	Working Draft (Borrador de Trabajo)	Borrador preliminar para su debate por un grupo de trabajo
3	CD	Committee Draft (Borrador de comité)	Borrador completo para votación y comentarios técnicos por cuerpos nacionales
4	CD,TR o TS	Committee Draft, Technical Report/Specification (Borrador de comité de un informe o especificación técnica)	Borrador final para votación y comentarios editoriales por cuerpos nacionales
	CDV	Committee Draft for Vote (Borrador de comité para votación (IEC))	
	DIS	Draft International Standard (Borrador de un estándar internacional)	
	FCD	Final Committee Draft (Borrador final de comité (JTC1))	



	DTR o DTS	Draft Technical Report/Specification (Borrador de una especificación o informe técnico)	
5	ISO	International Standard (Estándar internacional)	Texto para la publicación por aprobación final
6	ISO, TR o TS	Technical Report or Specification (Especificación o informe técnico)	Documento publicado

Algunos de los estándares más importantes, desarrollados por la organización ISO, que están relacionados con la IPO, se describen a continuación.

**ISO/IEC 9126 (Evaluación de productos software – características de calidad y directrices para su uso).**

Define la usabilidad como una contribución relativamente independiente a la calidad del software, asociada con el diseño y la evaluación de la interfaz de usuario y la interacción. Además define métricas para la usabilidad y la calidad de uso. El estándar se compone de las siguientes partes:

- Parte 1: Modelo de calidad.
- Parte 2: Métricas externas.
- Parte 3: Métricas internas.
- Parte 4: Métricas para la calidad de uso.

**ISO 9241 (Requisitos ergonómicos para trabajar con terminales de presentación visual (*Visual Display Terminals*, VDTs)).**

Este estándar se centró originalmente en requisitos hardware para los VDT, sin embargo evolucionó incluyendo también los requisitos software. Está compuesto de diecisiete partes que tratan cuatro áreas principales:

- Partes 1-2: Introducción y descripción general.
- Partes 3-9: Requisitos hardware.

- Parte 6: Requisitos del entorno.
- Partes 10-17: Requisitos software.

### **ISO/IEC 10741 (Interacción de diálogos).**

Define cómo deben iniciar, controlar y monitorizar los usuarios las posibilidades del sistema para controlar los cursores en sistemas de texto por medio de funciones de control. Consta de una única parte establecida desde 1995 que especifica cómo debería moverse el cursor sobre la pantalla en respuesta al uso de las teclas de los cursores.

- Parte 1: Control del cursor para la edición de textos.

### **ISO/IEC 11581 (Símbolos y funciones de los iconos).**

Define los datos y las funciones que el sistema realizará cuando el usuario interactúe con cierto tipo de iconos o imágenes. Consta de seis partes que detallan los iconos que se muestran por pantalla.

- Parte 1: Iconos en general.
- Parte 2: Iconos de objetos.
- Parte 3: Iconos de punteros.
- Parte 4: Iconos de controles.
- Parte 5: Iconos de herramientas.
- Parte 6: Iconos de acciones.

### **ISO 11064 (Diseño ergonómico de centros de control).**

Define los principios ergonómicos para diseñar centros de control y estaciones de trabajo. Consta de ocho partes.

- Parte 1: Principios para el diseño de centros de control.
- Parte 2: Principios para la organización del control.
- Parte 3: Disposición del sitio de control.
- Parte 4: Disposición y dimensionamiento de las estaciones de trabajo.
- Parte 5: Interfaces persona-sistema.

- Parte 6: Requisitos del entorno para los sitios de control.
- Parte 7: Principios para la evaluación de los centros de control.
- Parte 8: Requisitos ergonómicos para aplicaciones específicas.

### **ISO 13406 (Requisitos ergonómicos para trabajar con presentaciones visuales basadas en paneles planos).**

Se establecen los requisitos ergonómicos de la calidad de la imagen para el diseño y evaluación para presentaciones en 2D así como los métodos para determinar la calidad de la imagen. Consta de dos partes.

- Parte 1: Introducción.
- Parte 2: Requisitos ergonómicos para presentaciones visuales en paneles planos.

### **ISO/IEC 14754 (Gestos comunes para la edición de textos con sistemas basados en lápiz).**

Define el conjunto de comandos de gestos básicos y reacciones para interfaces basadas en lápiz que incluye: seleccionar, borrar, insertar espacio, saltar línea, mover, copiar, cortar, pegar desplazar y deshacer.

- Parte 1: Gestos comunes para la edición de textos con sistemas basados en lápiz.

### **ISO/IEC 15910 (Proceso de documentación de software de usuario).**

Define el proceso mínimo para crear la documentación del software con una interfaz de usuario. Incluye las pautas para crear la documentación impresa, los manuales de usuario y la referencia rápida así como la documentación en línea

- Parte 1: Proceso de documentación de software de usuario.

### **ISO 13407 (Procesos de diseño centrados en la persona para sistemas interactivos).**

Proporciona una orientación sobre las actividades de diseño centradas en la persona a lo largo del ciclo de vida de sistemas interactivos basados en ordenadores. Define el diseño como una actividad multidisciplinar que incorpora factores humanos y técnicas ergonómicas

con el fin de conseguir efectividad, eficiencia y mejora en las condiciones de trabajo de las personas.

- Parte 1: Procesos de diseño centrados en la persona para sistemas interactivos.

### **ISO/IEC 14598 (Evaluación de productos software).**

Define la calidad de uso y la forma de evaluar una aplicación software. Se compone de seis partes.

- Parte 1: Visión general.
- Parte 2: Planificación general.
- Parte 3: Proceso para los desarrolladores.
- Parte 4: Proceso para los adquirientes.
- Parte 5: Proceso para los evaluadores.
- Parte 6: Documentación de los módulos de evaluación.

### **ISO TR 18529 (Descripciones de los procesos del ciclo de vida centrados en la persona)**

Define la evaluación de un diseño centrado en la persona. Incluye una lista formalizada y estructurada de procesos centrados en la persona: especificar el usuario y los requisitos de la organización, producir las soluciones de diseño, evaluar los diseños a partir de los requisitos, etc.

- Parte 1 Descripciones de los procesos del ciclo de vida centrados en la persona.

### **ISO 10075 (Principios ergonómicos relacionados a la sobrecarga mental).**

Define los principios de diseño para aplicaciones que no produzcan una sobrecarga mental del usuario.

- Parte 1: Términos y definiciones generales.
- Parte 2: Principios de diseño.

**ISO 14915 (Ergonomía del software para interfaces de usuario multimedia).**

Describe las técnicas ergonómicas normalizadas para el diseño de interfaces de usuario eficaces.

### 3.3) Guías de estilo

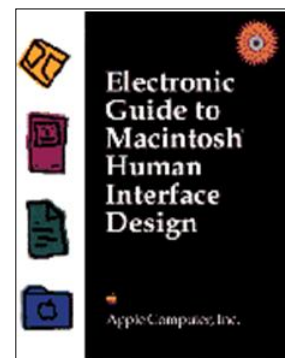
Un sistema interactivo consistente y robusto, requiere de los principios y directrices del diseño de interfaces. Además si este sistema se incorpora dentro de otro sistema operativo o aplicación, es necesario mantener la filosofía donde se enmarca. Por este motivo, las organizaciones proporcionan las guías de estilo, dirigidas a los desarrolladores de sistemas.

Se pueden dividir en: guías de estilo comerciales, también consideradas como estándares “de facto” y producidas por fabricantes de software y hardware; y las guías de estilo corporativas que elaboran empresas para su propio uso.

#### 3.3.1) Guías de estilo comerciales

Diseñadas por empresas dedicadas al software, contienen directrices a bajo nivel. Las más importantes están basadas en los sistemas operativos (“OS” *Operative Systems*) más comercializados. Algunas de ellas son:

“Apple” Publicó su primera guía de estilos en 1985 “*Human Interface Guidelines: The Apple Desktop Interface*” Posteriormente en 1992 publicó una nueva “*Macintosh Human Interface Guidelines*”, que contenía ejemplos animados para demostrar el empleo correcto de los elementos de interfaz de Macintosh, el comportamiento de los menús, los diálogos, etc. La última publicación ha sido “*Mac OS 8 Human Interface Guidelines*” que describe las incorporaciones y los cambios a realizar para la nueva versión 8 del Mac OS.



**Figura 46: Guía de estilo de Apple Corporation**

“IBM” publicó en 1987 la primera guía de normas CUA “*Common User Access*”. La segunda versión aparecería en 1989 con el nombre de “*Advanced Interface Design Guide*”.



Figura 47: Guía de estilo IBM (1992)

Estas normas tuvieron mucho éxito con los sistemas operativos DOS/Windows y OS/2. En el año 1992, IBM ya separada de Microsoft, publica “*Object-Oriented Interface Design: IBM Common User Access Guidelines*” donde se documenta completamente uno de los primeros entornos de trabajo orientados a objetos.

“Microsoft” publicó su primera guía en solitario a comienzos del año 1992, “*Microsoft. The Windows Interface: An Application Design Guide*”. Posteriormente, en el año 1995 publica la segunda guía de estilos, “*The Windows Interface Guidelines for Software Design*” que cubría el diseño de interfaces para Windows 95. Por último, en el año 2000, publicaría “*The Microsoft Windows User Experience*” que establecía los principios para el diseño de interfaces en Windows 98 y 2000. Además también incluía recomendaciones para el diseño de aplicaciones Web.

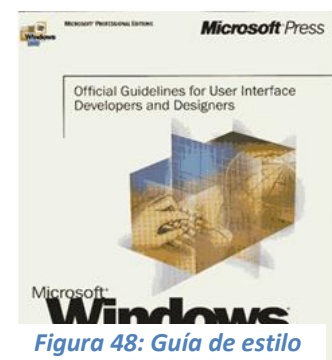


Figura 48: Guía de estilo Microsoft (2000)

Existen multitud de guías de estilo realizadas por otros fabricantes de software como: CDE (*Common Desktop Environment*); *Motif*; *Open Look*; *Java Look and Feel*; etc.



Figura 49: Guía de estilo Motif (Izquierda) y Java (Derecha)

### 3.3.2) Guías de estilo corporativas

Se trata de directrices que han de seguir todos los productos de una misma compañía. El objetivo de estas guías es mantener la identidad corporativa mediante gráficos, iconos, colores, logotipos de compañía, etc.

Para crear una guía de estilo de un producto o conjunto de productos, hay que seguir una serie de consideraciones. Generalmente los productos desarrollados se enmarcarán dentro de un sistema operativo ya creado. Por este motivo, tendrá que cumplir gran parte de las recomendaciones de las guías de estilo comerciales de este sistema operativo.

Por otro lado, dentro del contexto nacional e internacional, como ya se ha visto anteriormente (apartado 3.2, Cap. 3), existen una serie de estándares a los que habrá que ceñirse al diseñar la guía corporativa.

Según lo expuesto, para la creación de una guía de estilo de un producto, habrá que seguir la siguiente pirámide:



*Figura 50: Proceso para la creación de una Guía de estilo de un producto. Se comenzará desde abajo y se irá ascendiendo por la pirámide cumpliendo cada fase.*

## 3.4 Conclusiones

Ante el imparable desarrollo de las tecnologías y la aparición de numerosos sistemas interactivos, es necesario establecer unas pautas claras que guíen a los desarrolladores en su labor de diseño y que se pongan del lado del usuario para ayudarlo en el uso de las aplicaciones.

Los desarrolladores ven facilitado su trabajo al disponer de normas y estándares basados en estudios de alto nivel, y disponen de guías para desarrollar interfaces en cualquier sistema operativo o entorno de programación.

Así mismo el usuario, se encontrará con aplicaciones que siguen la misma filosofía de uso, facilitando así su aprendizaje y aprovechando el conocimiento previamente adquirido con otros productos.



# ***CAPÍTULO 4***

## **LA HERRAMIENTA DE MECÁNICA DE MÁQUINAS**

---

# I) ANÁLISIS DE REQUISITOS

## 1.1) La Asignatura de Mecánica de Máquinas

Los objetivos de la asignatura tienen como fondo una idea central: formar al alumno en la disciplina de Ingeniero mecánico. A su vez estos objetivos se dividen en dos, según sean de carácter general, o de carácter específico (referidos a conocimientos más técnicos).

Este módulo se centrará en el segundo tipo de objetivos, aquellos que proporcionan los conocimientos teóricos y prácticos específicos, que el alumno necesitará para el ejercicio de la profesión de Ingeniero Industrial. Según Cristina Castejón (2005) [43] son los siguientes:

- La comprensión y el dominio del lenguaje específico de la disciplina. Poder reconocer cada uno de los componentes y sistemas de una máquina real.
- La aplicación de los conocimientos básicos de mecánica del sólido rígido al estudio de sistemas multicuerpo empleados en sistemas mecánicos reales.
- El dominio de las técnicas gráficas de análisis cinemático y dinámico de mecanismos planos. El conocimiento y aplicación de las técnicas analíticas.
- El conocimiento de los aspectos topológicos, cinemáticos y dinámicos de los sistemas mecánicos.
- La familiarización con las técnicas de análisis y síntesis de los elementos constitutivos de las máquinas que posibiliten la formación de criterios de valoración de procesos y equipos, así como el posterior aprendizaje de actualización de los conocimientos adquiridos.
- La obtención de un grado de preparación suficiente que permita abordar las asignaturas del bloque de la especialidad y de los estudios posteriores de doctorado y/o máster, para así dirigir el área del análisis, síntesis y diseño mecánico de una empresa.
- Asentamiento de los principios fundamentales que rigen el estudio de los elementos de máquinas.
- Presentar el ámbito de utilización del ordenador en el diseño y mantenimiento de máquinas.

La asignatura está compuesta por siete bloques temáticos, para los cuales es necesario haber cursado y tener una formación previa en asignaturas del primer curso: Física I y II, Cálculo I y II, Álgebra Lineal I y II.

La herramienta que propone este proyecto solo abarcaría los ejercicios de los tres últimos bloques, los cuales se presentan a continuación:

- Bloque V – Mecanismos planos.
  - Introducción.
  - Partes constitutivas de un mecanismo.
  - Movilidad de un mecanismo.
  - Cuadrilátero articulado.
  - Determinación de los CIR relativos.
  
- Bloque VI – Cinemática de mecanismos planos.
  - Determinación de velocidades en miembros de un mecanismo.
  - Determinación de aceleraciones en miembros de un mecanismo.
  - Relación de aceleraciones y velocidades de puntos de pares cinemáticos.
  - Cinema de velocidades.
  - Cinema de aceleraciones.
  
- Bloque VII – Dinámica de mecanismos planos.
  - Introducción.
  - Análisis cineto-estático de mecanismos planos.
  - Análisis estático.
  - Análisis de los esfuerzos de Inercia.
  - Análisis dinámico completo.

El temario de la asignatura se imparte en tres tipos de clases: clases teóricas, donde se desarrollan los aspectos conceptuales; clases de ejercicios, para practicar y fijar conocimientos de teoría; y prácticas de laboratorio, donde se desarrollan los conceptos aprendidos en clase.

La aplicación de Mecánica de Máquinas tiene como objetivo reforzar las clases de ejercicios de la asignatura. Sin embargo, esto no implica la sustitución de ninguno de los

bloques que abarca, ni pretende eliminar las horas destinadas a la asistencia a las clases de ejercicios.

Para la utilización de la herramienta se requieren los conocimientos teóricos de los cuatro primeros bloques, así como los conceptos de los tres últimos. Pese a que la propia aplicación contiene detalladamente la resolución de los ejercicios (y por tanto los conceptos teóricos de los bloques que abarca), la labor del profesor no será relevada en ningún caso, manteniendo éste su papel en el proceso de aprendizaje del alumno.

El objetivo último de la herramienta es facilitar el proceso de aprendizaje de la asignatura. Como se verá más adelante (apartado 1.2, Cap.4), la asignatura de mecánica de máquinas posee cierta complejidad a la hora de resolver ejercicios gráficamente. Aprovechando el gran auge de las NTIC y el potencial de las herramientas de diseño gráfico y programación, se pretende explicar este tipo de problemas de la forma más sencilla y visual posible.

## 1.2) Dificultades de la asignatura

### El bloque V

Consistirá en el cálculo de los Centros Instantáneos de Rotación (CIR) mediante diversos teoremas. Para resolver los problemas se utilizará el teorema de Kennedy.

El cálculo de los CIR es un proceso que sigue generalmente unos pasos concretos. Como define en la referencia [44], los pasos son los siguientes:

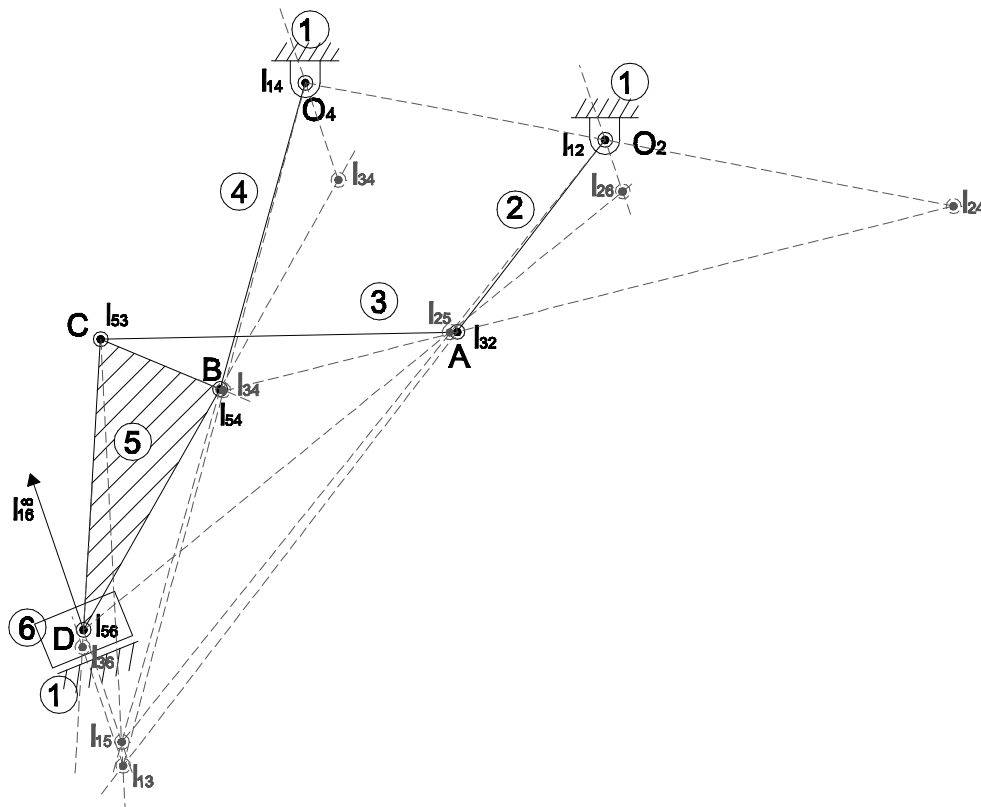
- 1. Determinar el número de eslabones del mecanismo.
- 2. Calcular el número de CIR's presentes: para ello se utiliza la fórmula combinatoria, que relaciona todos los eslabones entre sí dos a dos, donde N es el número de eslabones:

$$n_{\text{CIR}} = C_N^2 = \binom{N}{2} = \frac{N \cdot (N-1)}{2}$$

- 3. Determinar los CIR inmediatos: algunos centros, debido a su propiedad de velocidad nula, son sencillos de determinar, simplemente observando el mecanismo. Es el caso de manivelas y puntos de unión en pares cinemáticos de rotación y traslación.
- 4. Por último, determinar los CIR restantes aplicando el teorema de los tres centros, o teorema de Kennedy.

La dificultad que presentan estos problemas es la representación gráfica necesaria para resolver el ejercicio mediante el teorema de Kennedy. La explicación de este teorema mediante texto estático (como se viene haciendo hasta ahora en la asignatura) resulta menos visual y más compleja que si se realizara mediante una aplicación interactiva. Al ver la estructura ya resuelta (Figura 51), con los CIR dibujados, es imposible apreciar el proceso seguido para llegar hasta la solución. Por otro lado, si se utilizase una figura por cada paso, se requeriría gran cantidad de espacio y sería difícil de seguir.

Lo más efectivo para explicar cada paso, en el cálculo de los CIR, es mediante una aplicación interactiva que permita al usuario navegar, repetir, parar y en general interactuar con el proceso de explicación.



*Figura 51: Representación gráfica de los CIR de un mecanismo [44]*

## El bloque VI

Introduce el análisis de mecanismos articulados. Aborda el estudio de la cinemática de los mecanismos, presentando los métodos analíticos y gráficos. Los ejercicios consistirán en el cálculo, mediante técnicas gráficas, de las velocidades y aceleraciones de los puntos pertenecientes al mecanismo en un instante dado.

En este bloque por lo tanto existen dos tipos de problemas que se resolverán a través de la herramienta. Los relacionados con el cinema de velocidades y el de aceleraciones.

Para el **cinema de velocidades**, se presentan varios métodos para resolver los ejercicios gráficamente. Se utilizará el método de los puntos homólogos en la aplicación. El Cinema de velocidades es el lugar geométrico de los extremos de los vectores velocidad de todos los puntos de los eslabones de un mecanismo. Es más sencillo pasar desde el eslabón del mecanismo al cinema y viceversa utilizando las propiedades geométricas de la homología (conservación de ángulos y proporciones).

En este tipo de ejercicios la representación gráfica del cinema es relevante. El ejercicio se resolverá eslabón por eslabón, incluyendo en el cinema toda la información que se tenga de ellos (el módulo, la dirección o el sentido de la velocidad). Los vectores de velocidad absoluta partirán del polo de velocidades (un punto arbitrario denominado “O”) Los vectores de velocidades relativas se colocarán a continuación de los vectores de velocidad absoluta. Donde intersecan los vectores, se obtendrán los puntos homólogos de las uniones del mecanismo original. Se puede encontrar toda la teoría referente al cálculo de velocidades mediante el cinema de velocidades en la referencia ([44], apartado 3.3)

Una de las propiedades del cinema de velocidades es que la longitud de sus vectores es proporcional al valor real de su módulo, por lo tanto se pueden obtener las velocidades midiendo directamente del cinema.

Como ocurría con los CIRs, presentar en una figura estática el cinema de velocidades no resulta fácil de entender en muchos casos (Figura 52). Por otro lado el uso de muchas imágenes para explicar cada paso ocuparía gran cantidad de espacio y podría resultar confuso. La aplicación interactiva mostrará al alumno mediante figuras dinámicas y transiciones de elementos, cómo se forma el cinema paso por paso. Se pretende eliminar las explicaciones con términos geométricos y obtener el resultado con figuras y gráficos más intuitivos y visuales.

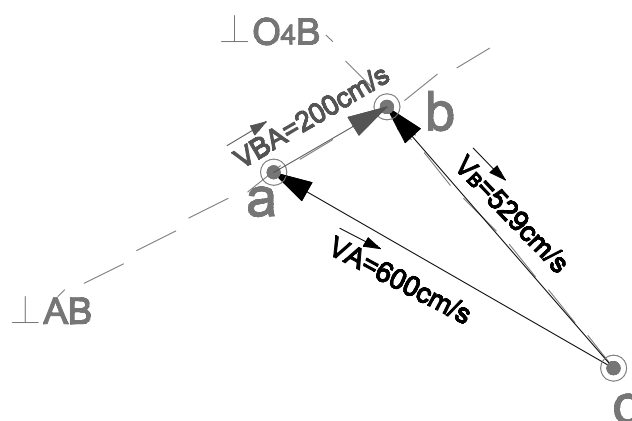
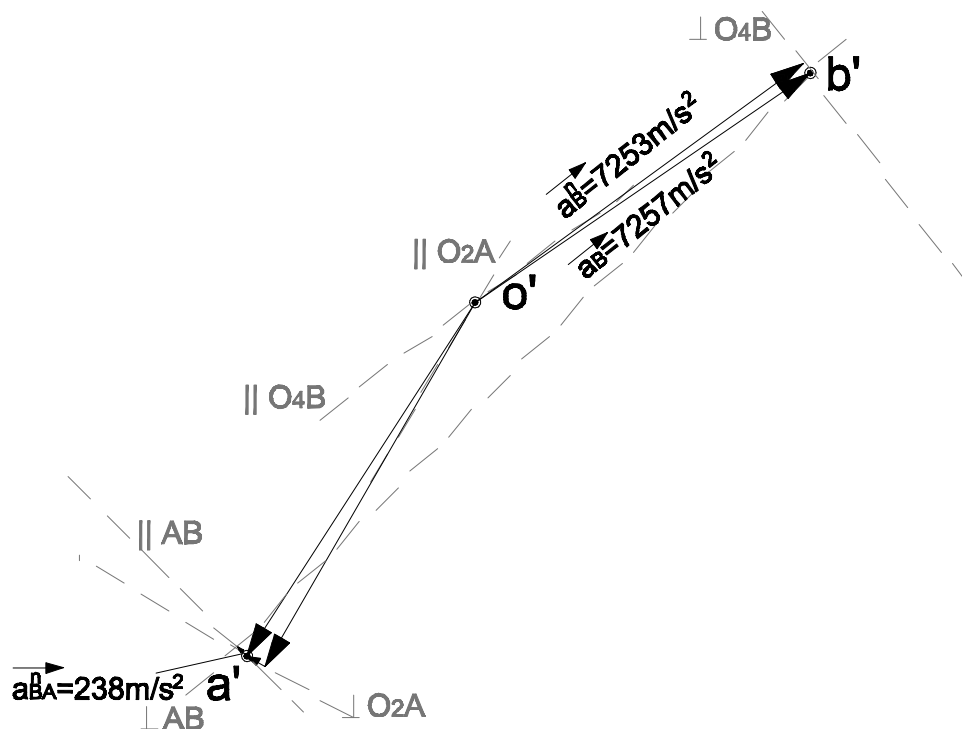


Figura 52: Obtención de los vectores velocidad de una estructura a partir del Cinema de Velocidades [44]

Por otro lado, el tema de **cinema de aceleraciones** aborda el estudio de las componentes de la aceleración en un punto del mecanismo, conocida la velocidad. Este estudio es semejante al cinema de velocidades, teniendo en cuenta que la aceleración de un punto del eslabón se dividirá en dos componentes: la aceleración normal (paralela al eslabón) y la tangencial (perpendicular al eslabón). Por analogía se presenta el concepto de polo de aceleraciones y de punto homólogo del cinema de aceleraciones.

Al igual que en el cinema de velocidades, el de aceleraciones se irá construyendo poco a poco con la información de cada eslabón. Se puede encontrar toda la teoría referente al cálculo de aceleraciones mediante el cinema de aceleraciones en la referencia ([44], apartado 3.4)

En este caso, el uso de figuras dinámicas y visuales adquiere más importancia, ya que el número de vectores utilizados se multiplica y resulta en ocasiones caótico descifrar el Cinema (Figura 53).



*Figura 53: Determinación gráfica de la aceleración de un punto [44]*

## El bloque VII

En el bloque VII se estudian los esfuerzos internos en los pares cinemáticos producidos por el movimiento, así como los esfuerzos exteriores sobre el mecanismo que generan este movimiento. ([44] Capítulo 4)

Este bloque se divide en dos partes:

- La primera consistirá en el cálculo de la Fuerza Reducida de un sistema y la Fuerza Equilibrante. La fuerza reducida, aplicada en un extremo del mecanismo y con dirección perpendicular a la dirección del eslabón, representa el conjunto de todas las fuerzas externas sobre el mecanismo (cumpliendo el Principio de los Trabajos Virtuales, PTV). La fuerza equilibrante será igual a la fuerza reducida pero en sentido opuesto, con el fin de anularla y mantener el mecanismo en equilibrio.

Para la obtención de la fuerza reducida de un mecanismo sometido a fuerzas exteriores, se pueden aplicar dos métodos:

- Método gráfico de reducción de fuerzas: se trata de un método sencillo, pero muy visual. Consiste en desplazar los vectores de las fuerzas aplicadas, haciéndoles coincidir con las direcciones de los eslabones. A continuación se descomponen en estas direcciones y se desplazan hasta el siguiente nodo. Las fuerzas que recaen sobre los apoyos se anulan. (Figura 54)

Al igual que en los cinemas, es difícil enseñar el método con figuras estáticas. Mostrando el resultado final resulta complicado seguir el proceso. Poniendo figuras paso por paso, se emplearía una gran cantidad de espacio para un resultado poco eficiente. El sistema más eficaz es mediante figuras dinámicas donde el usuario pueda ver el movimiento de los vectores deslizantes y seguir con facilidad el proceso. Este es el método que se usará en la aplicación de Mecánica de Máquinas.

- Método del Principio de los Trabajos Virtuales (PTV): para la obtención de la fuerza reducida por este método, es necesario aplicar una fórmula que relaciona la fuerza sobre el eslabón y la velocidad del punto donde se aplica dicha fuerza:

$$\sum_{i=1}^3 \vec{F}_i \cdot \vec{v}_{pi} + \vec{E}_{nodo} \cdot \vec{v}_{nodo} = 0$$

Donde:

$\vec{F}_i \rightarrow$ Fuerza aplicada en el eslabón i.	$\vec{E}_{nodo} \rightarrow$ Fuerza equilibrante en el último nodo.
$\vec{v}_{pi} \rightarrow$ Velocidad en el punto “p” de aplicación de la fuerza.	$\vec{v}_{nodo} \rightarrow$ Velocidad en el último nodo.

La única parte que puede presentar dificultades a la hora de seguir el proceso es el cálculo de las velocidades. Sin embargo, se emplean los métodos de los cinemas de velocidades ya explicados en el bloque anterior. Por lo tanto este método no se utilizará en la resolución de los ejercicios de reducción de fuerzas implementados para la aplicación.



- La segunda parte consiste en el cálculo de los esfuerzos de inercia en un mecanismo. El método de resolución también presenta ciertas dificultades visuales. En este caso, se estudia eslabón por eslabón, aislándolos del resto. Al aplicarles una aceleración, aparecen, en el centro de gravedad, una fuerza y un momento de inercia. Para calcularlos será necesario saber las aceleraciones de ambos extremos del eslabón y aplicar las siguientes fórmulas:

$$\vec{F}_i = -m_i \cdot \vec{a}_{Gi} ; \vec{M}_i = -I_{Gi} \cdot \vec{\alpha}_i ; h_i = M_i/F_i$$

Donde:

$\vec{F}_i \rightarrow$ Fuerza de Inercia del eslabón i.	$\vec{M}_i \rightarrow$ Momento de inercia del eslabón i.
$m_i \rightarrow$ Masa del eslabón i.	$\vec{\alpha}_i \rightarrow$ Aceleración angular del eslabón i.
$\vec{a}_{Gi} \rightarrow$ Aceleración lineal en el CDG del eslabón i.	$I_{Gi} \rightarrow$ Momento de inercia estático respecto al CDG.
$h_i \rightarrow$ Distancia perpendicular de la dirección de la fuerza al CDG.	$M_i/F_i \rightarrow$ Módulos del momento y la fuerza de inercia.

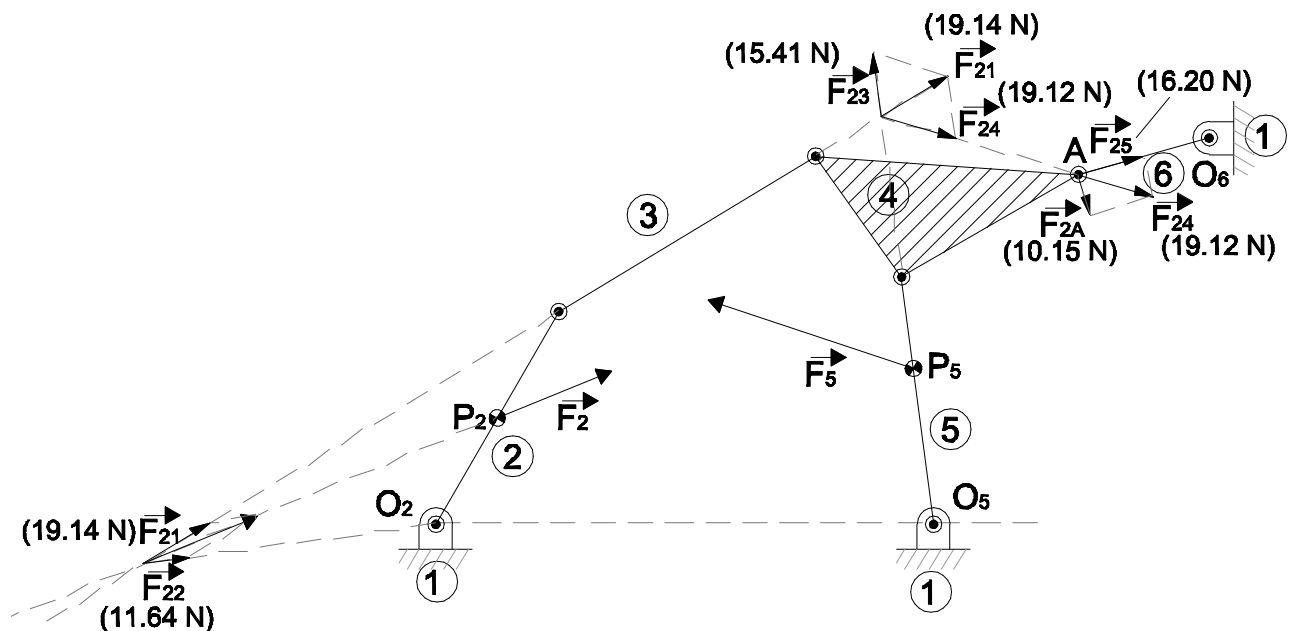


Figura 54: Ejemplo del método gráfico de reducción de fuerzas [44].

## 1.3) El alumno y su Entorno

Uno de los pasos previos al diseño de una aplicación interactiva es definir a quién va dirigida y cómo pensará el usuario de la misma. Como se explicó en el apartado de modelos mentales (apartado 2.3, Cap. 3), es necesario conocer el modelo mental del usuario para diseñar una aplicación y una interfaz entendible e intuitiva. El usuario ha de ser capaz de reconocer los elementos y darles el significado que el diseñador quiere.

La aplicación va dirigida a los alumnos de la asignatura de Mecánica de Máquinas. Se presupone un conocimiento de la materia (parte teórica) ya que la herramienta está diseñada como un apoyo a la realización de los ejercicios. El modelo mental que tendrá cada usuario sobre la asignatura y sus ejercicios, puede variar. Sin embargo se considerará que el usuario de la aplicación tiene un modelo mental correcto de la asignatura. Inculcar este modelo mental, será labor del profesorado.

En cuanto a la interfaz gráfica de la aplicación se usarán elementos tanto para la representación como para la navegación que se asumen conocidos por el usuario. La gran mayoría de alumnos que cursan esta asignatura, tendrán al menos un nivel medio de informática (se suponen cursadas asignaturas como programación) resultándoles los elementos utilizados “habituales” o “familiares”. Se emplearán las guías de estilo de los principales navegadores y la normativa existente para el diseño de la IU y de la interacción alumno-herramienta.

## II) DISEÑO

### 2.1) Descripción del funcionamiento

Según se explicó en el apartado de ciclo de vida de una aplicación interactiva (apartado 1.1, Cap. 3), en la etapa de diseño, el primer paso es analizar las tareas que va a realizar la aplicación y cómo las va a realizar. Así mismo es necesario establecer las interacciones usuario-sistema.

Se pretende que la herramienta de Mecánica de Máquinas sea una aplicación muy intuitiva y visual, dejando de lado una posible versatilidad que en este caso no beneficiaría al aprendizaje.

El fin de la aplicación es enseñar, mediante pasos claros, el procedimiento de resolución de ejercicios relacionados con la asignatura. Por este motivo, todas las decisiones que se tomen para su diseño han de tener este objetivo como principal.

En el apartado de dificultades de la asignatura (apartado 1.2, Cap. 4) se concluye que el procedimiento para resolver los modelos de ejercicios propuestos, siguen un patrón (con pequeñas modificaciones en algunos ejercicios). De esta forma, el procedimiento más efectivo, será mostrar al alumno dichos pasos mediante animaciones y transiciones animadas y visuales. Además será necesario dotar a la aplicación de un sistema de navegación que permita al alumno moverse entre los diferentes pasos y poder ajustar la velocidad de aprendizaje.

El contenido de cada paso será lo más reducido posible con el fin de que el alumno comprenda claramente el ejercicio a medida que va avanzando. Se ha tenido más en cuenta el conocimiento procedural frente al declarativo. Al ser una herramienta cuyo fin es el de resolver los problemas de la asignatura, es más importante conocer cómo realizar los ejercicios (procedural) frente a la teoría que tienen como trasfondo (declarativo). Sin embargo en las explicaciones se intenta que la representación no sea una “caja negra” donde solo se muestre el resultado final. El programa muestra una explicación fundamentada con la teoría de la asignatura.

La interacción alumno-herramienta ha de ser intuitiva y con ciertas limitaciones. El alumno ha de poder averiguar fácilmente, al enfrentarse por primera vez a la herramienta, cómo se navega por ella y cómo va a actuar. Se diseñará una navegación que permita avanzar, retroceder y volver a repetir los pasos. Además se podrá activar sin necesidad de que la explicación haya acabado. Esto dotará a la herramienta de mayor fluidez en caso de que el

alumno tenga un nivel de conocimiento más elevado. En el otro extremo, para favorecer a aquellos alumnos que les cueste asimilar la evolución del ejercicio, se permitirá detener la ejecución del paso y volver a reanudarla. Así se ralentizará la velocidad, dando más tiempo al alumno para entender la explicación. Se usarán botones que cumplan los estándares y las guías de estilo de los principales navegadores web, de tal forma que el usuario los reconozca a simple vista.

Por otro lado, al establecer una navegación entre diversos pasos, es necesario que el alumno sepa dónde se encuentra. Se diseñará una barra de estado que indique que el programa está resolviendo el ejercicio y que muestre el paso en el que se encuentra la resolución.

Han de mostrarse los resultados que se van obteniendo en cada paso y mantenerlos durante el resto de la resolución. De esta forma, se evita tener que volver a pasos anteriores para recuperarlos. Ante una larga lista de datos, el alumno ha de saber cuales se han calculado en ese momento, y cuales son de pasos anteriores. Se resaltarán los datos obtenidos en el paso recién ejecutado.

En la resolución de los ejercicios de velocidades y aceleraciones, el cinema adquiere una gran relevancia. Se diseñará la aplicación de tal forma que se vea claramente de dónde salen los vectores que lo forman y qué relaciones geométricas cumplen con la estructura (paralelismo o perpendicularidad). En ocasiones los cinemas formarán figuras grandes con gran cantidad de datos. Para esos casos se creará un botón especial que aumente el cinema en una ventana nueva para hacerlo más visible.

Con todo esto, al tratarse de una aplicación con una secuencia lineal, que permite al usuario trabajar a su ritmo mediante retroalimentación e intentando que su programación sea lo más sencilla posible, se clasificará la aplicación de Mecánica de Máquinas como un programa ramificado (apartado 2.2.1, Cap. 2). La desventaja de este tipo de programas es que implican un grado de flexibilidad menor y por tanto la individualización de la herramienta a cada alumno también será menor. El motivo de haber diseñado así este programa consiste en que su uso será de apoyo al profesorado y no intenta sustituirlo en la labor de enseñanza (Como es propio de un ITS).

## 2.2) Entorno de la Herramienta y Normativa

En este apartado se definirá el entorno donde se ejecutará la aplicación, y las normas y directrices aplicadas a su diseño.

Como se vio en el apartado de objetivos (apartado 1.2, Cap. 1) la aplicación pretende ser un apoyo didáctico online por lo que su ejecución será a través de la web. En el apartado de *Macromedia Flash* (apartado 2.2.2, Cap. 3) se presentó esta como una herramienta dedicada especialmente a la creación de aplicaciones web, con el inconveniente de necesitar un reproductor (o *plugin* en caso de la web) para visualizar sus archivos. Sin embargo, hoy en día prácticamente todos los navegadores poseen este *plugin* o permiten al usuario instalarlo y actualizarlo de forma casi automática.

El marco donde se ejecutará la aplicación de Mecánica de Máquinas es por tanto una página web, albergada en los servidores de la Universidad Carlos III de Madrid (Leganés). El dominio es el siguiente:

“<http://maqlab.uc3m.es/TMM/tmm.html>” y  
 “<http://maqlab.uc3m.es/TMM/MECMAQ.html>”.

Se trata del dominio donde se encuentra la antigua aplicación de Teoría de Máquinas y Mecanismos [45]. Se pretende mejorar los bloques que coinciden con los descritos en el apartado de la asignatura de mecánica de máquinas (apartado 1.1, Cap. 4) y conseguir así una página web más completa y actualizada.



Figura 55: Página web “<http://maqlab.uc3m.es/TMM/tmm.html>” [45] desde el navegador Mozilla Firefox

En cuanto a la normativa aplicada a la creación de la aplicación, hay que diferenciar entre los estándares de iure y los estándares de facto (o guías de estilo comerciales). La


definición detallada se puede encontrar en el apartado de normativa y estándares (apartado 3, Cap. 3).

Los estándares de iure aplicados, y su influencia en la creación de la aplicación, se enumeran a continuación.

- **ISO TR 18529.** Se tiene en cuenta a la hora de plantear el ciclo de vida de la aplicación en el que se basa este capítulo.
- **ISO 9241.** (Partes 10-17) donde se definen los requisitos software ergonómicos para trabajar con el PC.
- **ISO/IEC 11581.** Se tiene en cuenta para el uso de iconos de acciones y controles aplicados en la interfaz para la navegación
- **ISO/IEC 9126.** Para establecer los modelos de calidad y de evaluación de la aplicación
- **ISO/IEC 15910.** Para crear el manual de usuario de la aplicación.

En el apartado de estándares de iure para la IPO (apartado 3.2.1, Cap. 4) se puede encontrar más información sobre estas y otras normas de iure.

Por otro lado, el diseño de la aplicación no se basa en ningún estándar de facto o guía comercial en concreto, como las expuestas en el apartado de guías de estilo (apartado 3.3, Cap. 3). Sin embargo se han tenido en cuenta directrices que siguen algunas de ellas, así como recomendaciones de guías de estilo corporativas de los navegadores (apartado 3.3.2, Cap. 3). Algunos ejemplos son:

- Utilización de flechas para marcar la navegación por las páginas/pasos. “Flecha hacia la derecha” para avanzar; “flecha hacia la izquierda” para retroceder; “flecha en círculo” en sentido antihorario para rehacer. Este tipo de flechas lo usan la gran mayoría de los navegadores por lo que el usuario ya las conoce y las tiene como presupuestas gracias a su experiencia.
- Uso de expresiones conocidas por el usuario como [\(+Detalles\)](#) para mostrar más información sobre aquello a lo que acompañe. En este caso se usa para agrandar los cinemas. Además se muestra en color azul que significa que el texto tiene un hipervínculo (se abrirá una nueva ventana con el cinema detallado).
- Empleo del cursor  (selección de vínculo) para hacer saber al usuario que la región donde está situado el puntero puede ser presionada. (Guía de estilo de Microsoft [46]).
- Uso de animaciones de progreso con una barra rellenándose de izquierda a derecha. Indica al usuario que el sistema está trabajando.

Aplicando las normativas, directrices y recomendaciones expuestas anteriormente, se pretende que el usuario reconozca todos los elementos de la aplicación gracias a su experiencia anterior como usuario de navegadores, sistemas operativos y programas. Como se supuso en el apartado del alumno y su entorno (apartado 1.3, Cap. 4), el alumno tiene conocimientos informáticos suficientes para conocer estos elementos. Aún así se adjunta en un manual de utilización de la aplicación (Anexo I).

## 2.3) Diseño e Implementación

En este apartado se definirá el estilo de la aplicación y se explicará cómo ha sido su implementación a través de la herramienta de autor *Adobe Macromedia Flash*. Observando el ciclo de vida de una aplicación, la implementación corresponde con el último paso antes del lanzamiento y el análisis del estilo pertenece a la etapa de diseño (apartado 1.1, Cap. 3). Aquí se explicarán las dos etapas juntas con el fin de ejemplificar las ideas del diseño con el producto final.

Por otro lado, el apartado se dividirá en dos partes: la referente al diseño e implementación de la Interfaz de Usuario (IU) y la referente a la navegación a través de la aplicación.

### 2.3.1) Diseño de la IU

Para estructurar el diseño de una Interfaz de Usuario eficiente es necesario seguir una estrategia de diseño (apartado 1.4, Cap. 3). Las tres partes de las que se compone una IU son: los mecanismos de interacción, la gestión de entradas del usuario y la presentación de la interfaz.

#### Mecanismos de Interacción

La interacción entre usuario y aplicación, está enfocada exclusivamente a la navegación a través de los pasos para resolver cada problema. El diseño de la navegación se explicará en el apartado siguiente (apartado 2.3.2, Cap. 4). Cabe destacar que los mecanismos de interacción empleados, son de dos tipos y se explican a continuación:



- Mecanismo de posicionamiento (apartado 1.4.1.1, Cap. 3). Se realizará utilizando los ejes de la pantalla en 2D con una realimentación espacial del cursor en pantalla.
- Mecanismos de selección (apartado 1.4.1.4, Cap. 3). Selección sobre un conjunto de botones organizados en una región o “mesa” donde se disponen con una presentación icónica y una ordenación según su función (“siguiente paso” a la derecha, “anterior paso” a la izquierda, y los controles de reproducción en medio)



*Figura 56: Representación de los botones dispuestos sobre una “mesa”, ordenados y con presentación icónica*

## Gestión de Entradas del Usuario

La aplicación recibe la información mediante un mecanismo de comunicación basado en eventos (*events*) (apartado 1.4.2, Cap. 3). Es el método más utilizado en aplicaciones interactivas. En este caso, el usuario tiene total libertad de interactuar con los botones de navegación, estableciendo, él mismo, el ritmo del aprendizaje. La aplicación responderá cuando se genere un evento ocasionado por una acción del usuario.

Solo se usarán dos tipos de eventos, y ambos serán eventos de entrada: mediante la posición instantánea del “mouse” (para cambiar entre los dos tipos de punteros:   dependiendo de donde esté situado) y mediante “click” del “mouse”.

La comunicación entre objetos se define como la acción que realiza la aplicación después del evento y se explicará más detalladamente en el apartado siguiente (apartado 2.3.2, Cap. 4).

## Presentación de la Interfaz

La etapa más importante en la estrategia de diseño de una IU es la implementación de la última capa, aquella que ve el usuario. Toda la información sobre su creación se encuentra en el apartado de diseño gráfico de la aplicación (apartado 1.4.3, Cap. 3). Los tres elementos clave que habrá que tener en cuenta, son: organizar, enfatizar y hacer reconocible la interfaz.

En cuanto a la **organización** se ha implementado una interfaz siguiendo la metáfora de “pizarras” en las que se presenta: el gráfico, la resolución y los datos obtenidos del problema (Figura 57).





*Figura 57: Interfaz de usuario de la aplicación de Mecánica de Máquinas (metáfora “Pizarras”)*

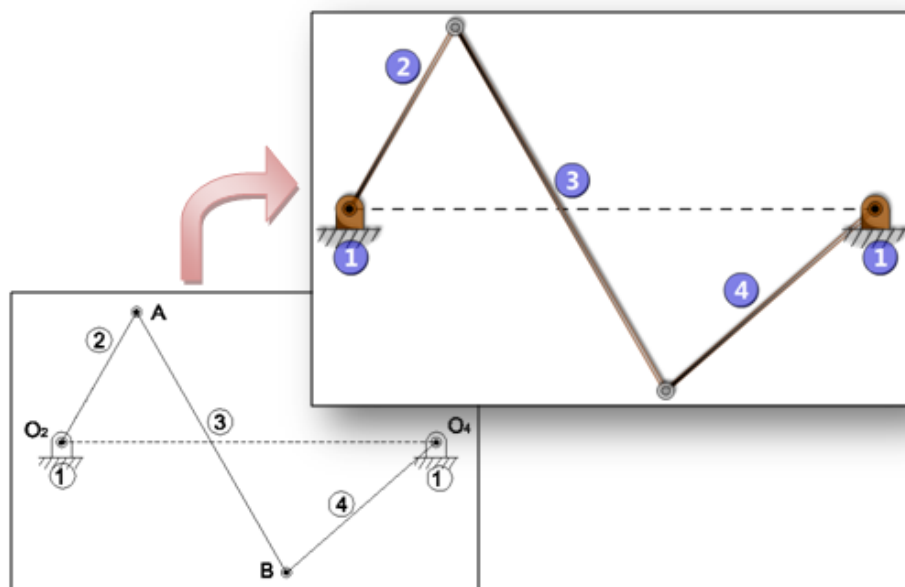
Después de varios prototipos (apartado 3, Cap. 4), ésta es la organización más eficiente y más clara para el alumno, teniendo en cuenta el contenido y la resolución de los problemas de la asignatura. La parte central la ocupará la pizarra con la estructura e inmediatamente debajo el panel de resolución. Estas serán las “pizarras” más importantes y donde más tiempo estará centrada la atención del usuario. A la derecha se presentarán los datos (en la “pizarra” más pequeña) y en la zona de abajo se disponen todos los botones necesarios para la navegación.

Para **enfatizar** la IU, se ha creado una jerarquía visual en la que destacan los tres paneles y los botones de navegación (más detalle en el apartado 2.3.2, Cap. 4). Además, durante la resolución de los problemas, se intenta captar la atención del alumno para guiarlo correctamente a través del flujo de los pasos. Para ello se realizan las siguientes animaciones gráficas:

- Aparición del texto mediante degradado según convenga para la resolución del paso.
- Aparición de objetos mediante degradado para seguir el flujo de resolución.
- Animación de vectores para que el alumno reconozca de donde vienen y donde han de colocarse en el cinema de velocidades.
- Animación de objetos mediante movimiento para transportarlos de un panel a otro (por ejemplo para llevar la solución del panel de resolución al de datos).
- Intensificación del color en los elementos que aparecen fuera del panel de resolución para llamar la atención visual del alumno y desviarla hacia donde corresponde.
- Aplicar transparencias a la figura del gráfico para resaltar aquello que interesa.

Para **hacer reconocible** la interfaz, se han incluido elementos icónicos comúnmente utilizados en otras aplicaciones y en las guías de estilo de los principales navegadores (flechas en el sistema de navegación) y se han eliminado todo tipo de elementos de identificación para hacer la interfaz lo más sencilla y minimalista posible. El alumno debe concentrarse en el problema y no en descifrar la interfaz.

En cuanto a los elementos utilizados durante la resolución, se considera que el alumno los conoce y los tiene interiorizados. Han debido ser enseñados con la teoría de la asignatura. Los vectores, expresiones y nomenclatura se han obtenido de la referencia [44]. Los elementos que componen las estructuras (apoyos, nudos, barras) se han modificado dándoles una apariencia más gráfica para mejorar la presentación de la interfaz (Figura 58).



*Figura 58: Nueva apariencia de los elementos de una estructura*

Para la creación del resto de la IU se han seguido algunas de las técnicas expuestas en el apartado de diseño gráfico de la aplicación. (apartado 1.4.3, Cap. 3):

- Reglas de Gestalt, en particular: la similitud entre botones con acciones de avanzar y retroceder (botones en verde) y los de intervenir en la ejecución del problema (botones en azul); cierre de áreas donde se representan elementos comunes delimitadas por las “pizarras” (gráfico, resolución y datos).
- Se han diseñado botones grandes siguiendo la ley de Fitt para que resulten fáciles de percibir y de pinchar para el usuario.
- Técnica de anticipación proporcionando indicaciones visuales de dónde ha de pinchar el usuario (cuando se termina el paso), asociando los iconos a las acciones de los botones y simplificando la interfaz.
- Se ha conseguido la autonomía de la IU creando una interfaz flexible pero con limitaciones en la navegación para evitar problemas o errores durante su ejecución. Además se incorporan dos indicadores de estado: barra de estado “resolviendo” e “indicador de paso”, para mantener al alumno informado a lo largo de la explicación.
- En cuanto a la percepción del color, se han seguido algunas de las pautas claves que se especifican en el apartado de diseño gráfico de la aplicación (apartado 1.4.3, Cap. 3): se ha usado una cantidad limitada de colores, uniformes y no llamativos; se utilizan los cambios de color e intensidad para captar la atención del usuario. Para resaltar el texto, se usa un gran contraste entre el color de las letras y el de las pizarras (negro sobre blanco).
- Se ha creado una aplicación consistente con estructuras visibles para la navegación que siguen el estilo de los navegadores web (flechas para ir a la página siguiente, retroceder a la anterior, rehacer/recargar la página web, pausar y reanudar la explicación; link para abrir la ventana del cinema detallado) y que el usuario interpretará y reconocerá rápidamente.
- Se busca la eficiencia del usuario eliminando todas las interferencias que pudieran interrumpir el proceso de aprendizaje. El elemento clave de la aplicación es la resolución del problema. Es al que más importancia se le ha querido dar para que el alumno pueda centrar en él toda su atención.
- Se ha diseñado una interfaz explorable con una navegación sencilla pero robusta. El alumno puede explorar la aplicación libremente sin peligro de ocasionar errores en la aplicación. Puede avanzar o retroceder a través de los pasos sin necesidad de completar su explicación. Se incluyen visualizadores de estado, para saber en cada momento dónde se encuentra.
- Se ha prestado mucha atención a la legibilidad de la IU debido a que ha sido uno de los principales problemas. Algunas explicaciones requerían gran cantidad de texto. La nomenclatura en las fórmulas, en los cinemas y en la estructura en ocasiones requería más espacio del disponible en las “pizarras”. La solución para obtener una IU legible consistió en:

- Dividir las explicaciones en más pasos, de tal forma que estos fuesen concisos, claros y bien estructurados.
  - Mostrar al alumno las explicaciones progresivamente para facilitarle la organización del texto.
  - Aumentar el tamaño de las “pizarras”.
  - Incorporar una ventana nueva donde se mostrase el cinema completo después de cada paso.
  - Aumentar el tamaño de letra para los símbolos y la nomenclatura del problema (tamaño de 12 puntos para el texto estándar, 14 puntos para simbología y nomenclatura).
- Se ha creado una interfaz visible, es decir, el alumno sentirá que se mantiene en un único lugar y que lo único que varía es su trabajo. Así se evita la necesidad de un mapa del contenido o ayudas de navegación y se consigue una mayor autonomía por parte del usuario.
- La realimentación del sistema es muy sencilla pero efectiva. Se usará una realimentación sobre la selección entre los botones de la navegación. Su dimensión temporal será: presente para la indicación de estado “resolviendo” y la indicación del paso en el que se encuentra el alumno. Pasada para la indicación de fin del ejercicio.
- No ha sido necesaria la gestión de posibles errores ya que la programación de la navegación es suficientemente robusta para no generarlos. Por otro lado el alumno no puede llevar a cabo acciones de no retorno por error o intencionadas que puedan inducir al fallo de la aplicación ya que se limita su interacción a lo realmente imprescindible.

En definitiva, este apartado se ha enfocado a conseguir un Diseño Centrado en el Usuario (DCU) definido por la norma ISO 13407. La motivación del alumno por usar la aplicación de Mecánica de Máquinas viene de la necesidad de entender la resolución de los problemas de dicha asignatura. Por lo tanto es vital que la aplicación satisfaga esa necesidad.

Para conseguir un diseño centrado en el usuario se ha basado el diseño de la interfaz en la percepción visual y en la cognición del alumno. La percepción visual, como se explicó en el apartado correspondiente (apartado 2.1, Cap. 3) se divide en varias fases. A continuación se explicará cómo percibe el alumno la aplicación de Mecánica de Máquinas desde que se inicia la resolución de un problema de la asignatura.

La primera percepción que tiene el alumno es masiva. El diseño de la interfaz hace que éste distinga cuatro partes diferenciadas. Los tres paneles o “pizarras” y la barra de botones abajo. Los elementos en su interior no serán focalizados. Después de haber asimilado la organización (nada más abrir la aplicación) la atención del alumno se desviará a la estructura, que por su uso de colores resultará visualmente más atrayente. Por último, se prestará atención al enunciado situado en la “pizarra” de resolución.

A medida que avanza en los pasos, el individuo mantendrá focalizada su atención en el panel de resolución, lugar donde se hace aparecer el texto explicativo en forma de degradado para llamar su atención. El resto de pizarras se encontrarán en la zona de visión periférica y el alumno no las focalizará.

En los pasos en los que sea necesario representar un vector, un CIR, una dirección o cualquier elemento fuera de la “pizarra” de resolución, se establece un punto de entrada que capte la atención del alumno (mediante un destello rojo, transparencia de elementos innecesarios, movimiento, etc.). Igualmente ocurre al terminar el paso, momento en el que se establece un punto de entrada en el botón de “siguiente paso”.

Así será la percepción del alumno, en la mayoría de los casos, cuando se enfrente por primera vez a la aplicación. Aún así, se ofrece la posibilidad de pausar la resolución del paso generando un punto muerto, sin animaciones, para permitir al alumno una exploración completa de la interfaz.

Por otro lado, se han tenido en cuenta las restricciones de la cognición humana (apartado 2.2, Cap. 3) para tomar las siguientes consideraciones:

- Interfaz simple ya que el alumno no será capaz de percibir más de 3 o 5 elementos.
- Conexión entre los pasos de la explicación debido a que el almacenamiento en la memoria de datos inconexo es mucho más complicado.
- Interfaz organizada y enfatizada para evitar pérdida de tiempo y de esfuerzo por parte del alumno.
- Prevención de errores para evitar que la toma de decisiones intuitiva desencadene en un fallo de la aplicación. Para ello se limita la acción del usuario exclusivamente a la navegación, se le orienta para que actúe correctamente y se le permite deshacer o rehacer las acciones llevadas a cabo.

Quedan explicadas de este modo, todas las decisiones tomadas para el diseño y creación de una interfaz de usuario para la aplicación de Mecánica de Máquinas.

### 2.3.2) Diseño de la navegación

El proceso de navegación a través de la aplicación de Mecánica de Máquinas es la única forma de interacción alumno-aplicación que se ha incorporado en la herramienta de aprendizaje.

Como se explicó en el apartado del funcionamiento (apartado 2.1, Cap. 4), la navegación ha de ser sencilla pero robusta. Este fue uno de los motivos por los que se seleccionó la herramienta de autor, *Macromedia Flash*. La navegación se programará con el lenguaje que incorpora la herramienta, *Action Script 3.0*.

En este apartado se describirán las principales funciones utilizadas para la programación de la navegación. El código específico de cada paso se podrá encontrar en el Anexo de metodología de creación de ejercicios (Anexo II).

La navegación se ha basado en el uso de botones con los que el usuario puede interactuar. El “botón” en Flash es un tipo de símbolo que al ser pulsado genera un evento que desencadena una acción (Figura 59):

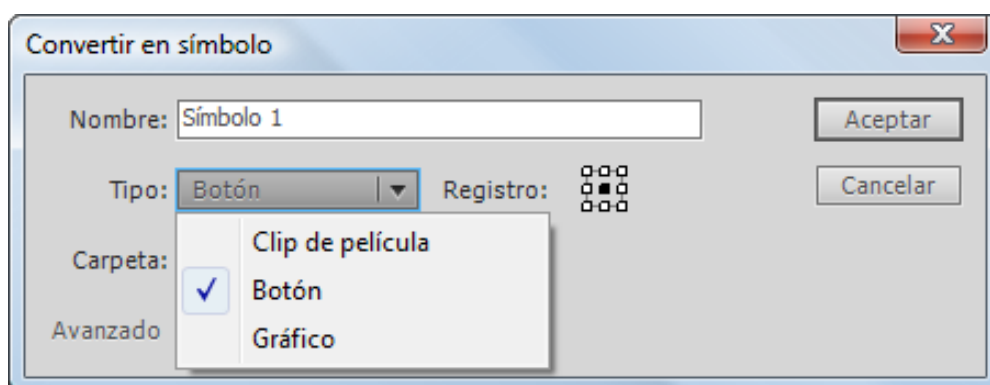


Figura 59: Tipos de símbolos en Macromedia Flash.

Los botones tienen sus propias características (“visible”, “enabled”, etc.) Además poseen una línea de tiempo propia que no se corresponde con la línea de tiempo de la animación. Solo posee cuatro fotogramas donde se muestran los cuatro posibles estados del botón: estado de reposo, estado de “ratón sobre el botón”, estado de “botón presionado” y la “zona del botón que desencadena la acción”. (Figura 60).

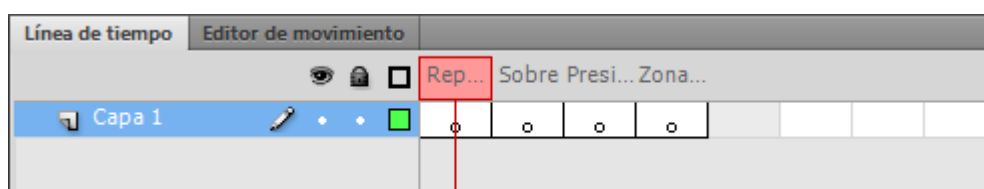


Figura 60: Línea de tiempo de un Botón.

Para entender la programación empleada en la navegación es importante definir los siguientes términos:

- “Event” o evento. Se trata de un mecanismo de comunicación del usuario con la aplicación (ver apartado 1.4.2, Cap. 3). La aplicación esperará a que ocurra un

evento para ejecutar una acción. El tipo de evento que se usará para la navegación es el “Evento de entrada” haciendo *click* sobre el botón.

- “*Event handler*”. Se trata del procedimiento que se lleva a cabo al producirse un evento.
- “*Callbacks*”. Es la vinculación del “*event handler*” al “*event*”. Se trata de un modelo de programación (modelo de llamadas) que usa *Action Script*.
- “*Event listener*” Se usa para convertir un símbolo (en este caso los botones) en un “objeto observador”, es decir, que *Acción Script* podrá recibir eventos de ese objeto.

```
siguiente_btn.addEventListener(MouseEvent.CLICK, IrPaso1);
```

En este ejemplo obtenido del código de la aplicación sometida a estudio, se distinguen los siguientes términos: “*siguiente\_btn*” es el nombre del botón; “*addEventListener*” hace del botón un objeto observador de eventos; “*MouseEvent.CLICK*” determina qué tipo de evento podrá darse en el botón, en este caso hacer *click*; “*IrPaso1*” es la función vinculada al evento, es decir, lo que se desencadena al pulsar en el botón. El “*Event handler*” será todo aquello que aparezca dentro de la función “*IrPaso1*”.

```
function IrPaso1(e:MouseEvent):void
{ gotoAndPlay("Paso1") }
```

Para implementar la navegación se ha creado una capa dedicada exclusivamente a la programación de la misma (capa llamada “*as*” de *actionscript*). La línea del tiempo de esta capa se ha estructurado en diferentes partes que determinan el inicio del paso, la resolución y el final (Figura 61). El fin de esta organización es facilitar la creación de nuevos ejercicios.

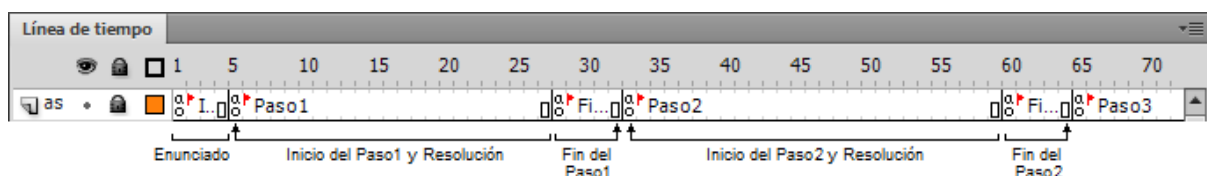







Figura 61: Estructura de la línea del tiempo de la capa “*as*”

El número de fotogramas para el enunciado y el fin de cada paso es invariable. La aplicación en estas fases será estática por lo que es indiferente el número de fotogramas que se usen. El único objetivo que tienen es hacer más visual la estructura y más fácil de interpretar para futuros desarrollos. Por otro lado, el número de fotogramas dedicados a la fase de “Inicio del Paso y Resolución” dependerá del contenido de cada paso.



El código se implementa en el primer fotograma de cada fase (fotogramas con una “a” inscrita) (ver Anexo II). En ellos se hará referencia a los siguientes botones con las propiedades que se especifican en la Tabla 6.


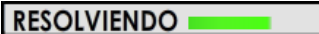
*Tabla 6: Elementos de la navegación.*

Botón	Nombre	Propiedades utilizadas	Funciones vinculadas
	siguiente_btn	<p>(Enunciado, resolución y fin de paso) Enabled = true Visible = true</p> <p>(Fin del último paso) Enabled = false Visible = false</p>	<pre>function IrPasoX(e:MouseEvent):void { gotoAndPlay("PasoX");}  (Donde X es el número de paso siguiente al actual)</pre>
	anterior_btn	<p>(Resolución y fin de paso) Enabled = true Visible = true</p> <p>(Enunciado) Enabled = false Visible = false</p>	<pre>function IrPasoX(e:MouseEvent):void { gotoAndPlay("PasoX");}  (Donde X es el número de paso anterior al actual)</pre>
	refresh_btn	<p>(Fin de paso) Enabled = true Visible = true</p> <p>(Enunciado y durante resolución) Enabled = false Visible = false</p>	<pre>function IrPasoX(e:MouseEvent):void { gotoAndPlay("PasoX");}  (Donde X es el número de paso actual)</pre>
	pause_btn	<p>(Durante resolución) Enabled = true Visible = true</p> <p>(Enunciado y Fin de paso) Enabled = false Visible = false</p>	<pre>function Pause(e:MouseEvent):void { stop(); progreso_mc.stop(); pause_btn.visible = false; play_btn.visible = true; }</pre>
	play_btn	<p>(Al pulsar Pause_btn) Enabled = true Visible = true</p> <p>(Al pulsar play_btn) Enabled = false Visible = false</p>	<pre>function Play(e:MouseEvent):void { pause_btn.visible = true; play_btn.visible = false; play(); progreso_mc.play(); }</pre>



Por otro lado, existen dos elementos que no pertenecen a la navegación como tal, pero que intervienen en su programación (Tabla 7).

*Tabla 7: Otros elementos que intervienen en la programación de la navegación.*

Elemento	Nombre	Descripción
	detalle_btn	Se trata de un botón cuyo estado al ser pulsado muestra una ventana con el cinema en detalle.
	progreso_mc	Se trata de una barra de progreso que indica que el paso se está resolviendo. Será visible durante la resolución del paso y desaparecerá cuando éste termine. Es un símbolo del tipo “ <i>movie clip</i> ” que posee una animación propia, independiente de la aplicación y que se reproduce en bucle.

En el primer fotograma de la línea de tiempo (que corresponde con el primer fotograma del enunciado, ver (Figura 61) se programarán todas las funciones o “*event holders*” a los que se llamarán desde los eventos que ocurran en el resto de pasos (*click* en los botones). Dependiendo del paso donde se encuentre el alumno, cada uno de los botones hará una función diferente. Por ese motivo, al comienzo de cada fase es necesario eliminar el “*event listener*” de cada botón y poner el nuevo que corresponda. A continuación se muestra un ejemplo del inicio del paso 1:

```
//Quitamos los listeners del enunciado
siguiente_btn.removeEventListener (MouseEvent.CLICK, IrPaso1);

//Añadimos los nuevos listeners para este paso
siguiente_btn.addEventListener (MouseEvent.CLICK, IrPaso2);
anterior_btn.addEventListener (MouseEvent.CLICK, IrInicio);
```

A continuación se muestra como ejemplo el fin del paso 1:

```
//Volvemos a quitar los listeners del enunciado
siguiente_btn.removeEventListener (MouseEvent.CLICK, IrPaso1);

//Añadimos el nuevo listener del botón refresh
refresh_btn.addEventListener (MouseEvent.CLICK, IrPaso1);
```

En ambos es necesario quitar los “*event listeners*” del enunciado (paso anterior). En el inicio del paso 1 añadimos como “*event listeners*” a los botones siguiente y anterior. En el fin del paso 1 solo añadimos el botón de *refresh*, puesto que los otros dos (siguiente y anterior) ya estaban añadidos. Los botones de *play* y *pause* siempre harán lo mismo

independientemente del paso que se esté resolviendo. Por lo tanto se añadirán como “*event listeners*” al comienzo del problema (en el primer fotograma).

Como se ha podido ver, la programación de la navegación está bien definida y estructurada, dotándola de sencillez y robustez. El alumno podrá explorar la aplicación intuitivamente y sin miedo a generar errores.

### III) PROTOTIPOS

La creación de la aplicación, como se vio en el apartado de diseño (apartado 2, Cap. 4) se divide en dos partes: la interfaz de usuario y la navegación. Desde el comienzo se han realizado prototipos de ambas partes por separado para estudiar su viabilidad y sus mejoras hasta llegar al resultado final que muestra este proyecto. Paralelamente al prototipado fue necesaria la evaluación de los mismos para ver sus puntos débiles y mejorarlos. Para ello se han seguido las pautas del apartado de evaluación (apartado 1.6, Cap. 3) así como algunas de las directrices establecidas por las normas ISO/IEC 9126 e ISO/IEC 14598

Se comenzó con la creación de la interfaz gráfica siguiendo todas las recomendaciones establecidas en el apartado de diseño de una aplicación interactiva (capítulo 3). Se crearon diversos diseños de prototipado horizontal (ver apartado 1.5, Cap. 3) en los que se buscaba la máxima semejanza al producto final sin incluir ningún tipo de navegación ni funcionalidad. A continuación se muestran los prototipos de alta fidelidad que fueron creados y cuáles fueron sus inconvenientes para ser descartados.

Para el primer prototipo se pensó en una ventana que mostrase diferentes secciones. Desde el principio se intentó buscar la idea de tener una sección dedicada a la estructura, otra a la resolución del problema, otra para los resultados y una pequeña barra de botones para la navegación (hasta el momento no implementada). Así se creó el primer prototipo mostrado en la (Figura 62).

Este primero prototipo utilizaba el estilo de las aplicaciones de “*Adobe*” ajustado a los requerimientos de la aplicación. Las ventajas que tenía eran: amplia región para la información y un aprovechamiento máximo del espacio. Sin embargo su principal problema fue la dificultad para distinguir a primera vista los diferentes paneles cuando contenían información. Se intentó dar más dinamismo a la aplicación para mejorar la monotonía en cuanto a percepción de color y organización de este primer prototipo.



Figura 62: Prototipo número 1 de la IU.

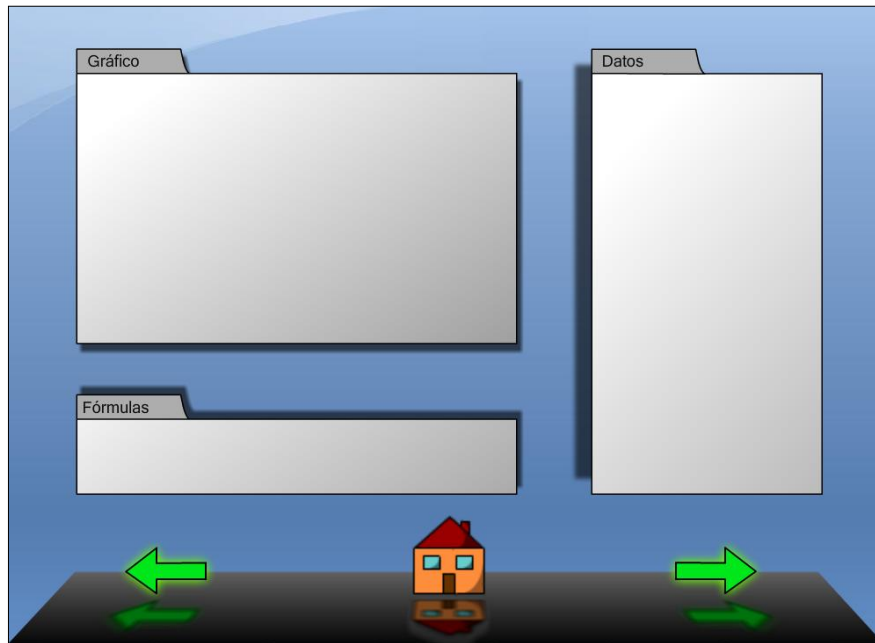
Con el segundo prototipo se mejoró la distinción de las secciones en la fase de percepción masiva del alumno (ver apartado 2.1, Cap. 3). Se apostó por unos paneles que sobresalían del fondo y con amplias separaciones entre ellos. Se seguían conservando las cuatro secciones (en este prototipo no se incorporaron los botones) (Figura 63).



Figura 63: Prototipo número 2 de la IU.

Este prototipo mejora la percepción de la organización, sin embargo la utilización de los colores y las formas no eran las apropiadas. Se buscaban colores más intensos para resaltar los paneles y más suaves para aquello que no tiene tanta importancia en la resolución del problema.

Siguiendo estas posibles mejores se diseña un tercer prototipo con colores más contrastados y unos botones más llamativos. Se incorpora una zona de botones utilizando la metáfora de una “mesa”, y se consigue dar la apariencia a los paneles de “pizarras” separadas del fondo. (Figura 64).



*Figura 64: Prototipo número 3 de la IU.*

Con este prototipo se consiguió resaltar la organización y la apariencia visual. Sin embargo los botones resultaban demasiado llamativos y el aprovechamiento del espacio era muy poco eficiente (apenas hay hueco para la resolución en el panel de “Fórmulas”).

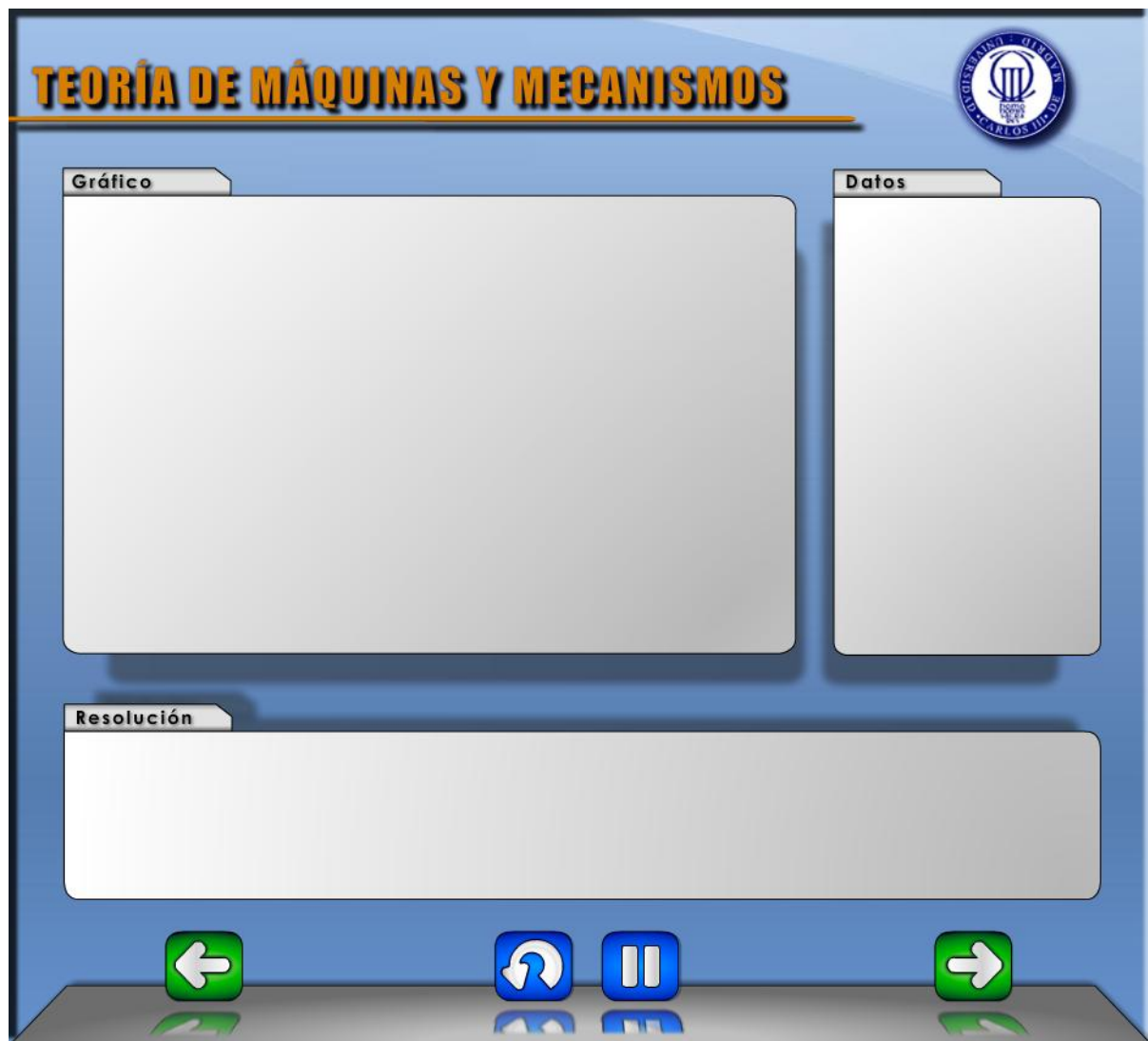
Se redujo el espacio para los datos, dando mayor amplitud al área de “Fórmulas” y se diseñaron unos botones con un aspecto visual menos llamativo pero más dinámico. Se incorporó el título y el logo de la universidad. (Figura 65).



*Figura 65: Prototipo número 4 de la IU*

Las mejoras para conseguir la plantilla final de la aplicación se realizaron sobre este prototipo. Se optimizó el espacio reduciendo el tamaño del logo y dejando menos distancia entre paneles para obtener un panel de “Fórmulas” más grande. Se cambió el nombre de este último panel por “resolución”. Se modificaron los botones y se crearon el resto, necesarios para la navegación.

Finalmente se crea una maqueta digital estática que incluye todos los elementos de la interfaz que se va a integrar en la aplicación. (Figura 66).



*Figura 66: Maqueta digital estática que representa la plantilla final.*

A continuación se explicará la evolución de la navegación y sus prototipos. En primer lugar se estudió el tipo de navegación que debería tener la aplicación y se obtuvieron las conclusiones que aparecen en el apartado de diseño de la navegación (apartado 2.3.2, Cap. 4). Posteriormente se realizaron diversos “*storyboards*” (ver apartado 1.5, Cap. 3) en papel para poder implementar un prototipo vertical (con botones prediseñados) de la navegación programado con *Action Script* (Figura 67).

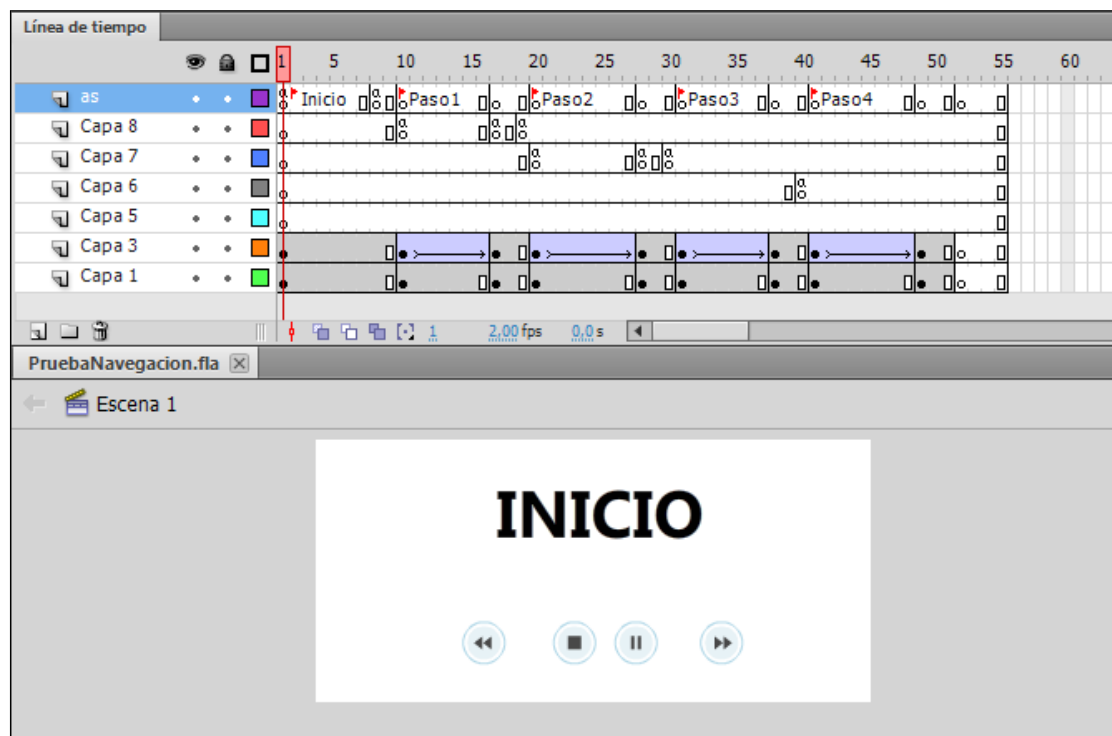


Figura 67: Prototipo número 1 de la navegación

A la hora de crear este primer prototipo se obtuvieron diversos problemas de prioridad alta en la programación (ver apartado 1.6, Cap. 3). Hasta conseguir el funcionamiento óptimo del prototipo se tuvieron que resolver errores en el código para poder continuar con el proceso de diseño. Algunos de los problemas fueron:

- Error al no eliminar los “*event listeners*” al final de cada fase. Generaban un funcionamiento impredecible de la navegación. Problema de nivel 4 considerado catástrofe (apartado 1.6, Cap. 3).
- Error al incorporar solo la propiedad “*visible*” de los botones. Estos desaparecían pero aún se podían seguir pinchando en la región donde se encontraban. Problema de nivel 2 con prioridad media. Se resolvió incorporando la propiedad “*enabled*” de los botones”.
- Problema al desaparecer todos los botones de la zona inferior cuando se programaba su visibilidad como “*false*”. Problema de nivel 1, estético de prioridad baja. Se solucionó añadiendo los mismos botones en forma de imagen con un cierto nivel de transparencia por detrás de los ya creados.

A continuación, se añadió el botón de “*refresh*” y se diseñaron los elementos interactivos de la Tabla 7. Finalmente se trasladó el código de este prototipo a la maqueta digital de la interfaz de usuario (explicada en este apartado), creando así la plantilla definitiva de la aplicación.



## IV) EVALUACIÓN

En este apartado se realizará un estudio de la aceptación de la aplicación entre diferentes usuarios con el fin de medir la calidad del software creado. Para ello se tiene en cuenta la norma ISO/IEC 9126 que se trata de un estándar para la evaluación de Software dividido en cuatro partes: modelo de calidad, métricas externas, métricas internas y calidad en las métricas de uso.

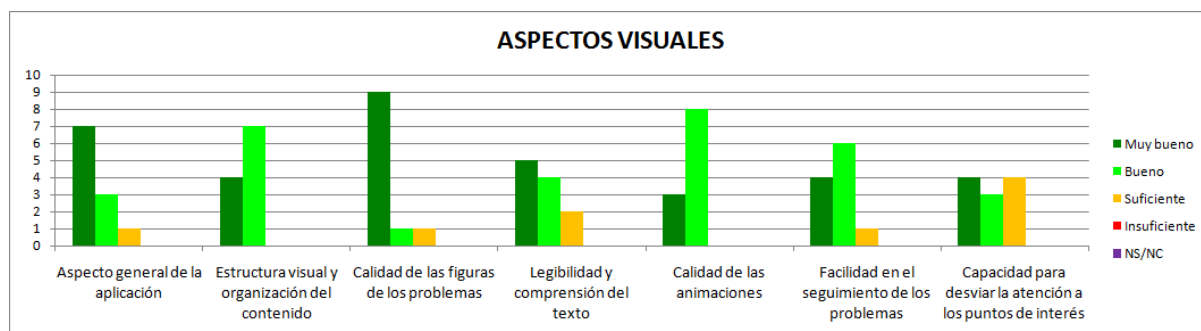
Se creará una matriz de evaluación basada únicamente en el modelo de calidad de la norma anterior. Así se establecerán tres aspectos a medir:

- Aspectos visuales: Con este apartado se pretende evaluar el impacto visual de la aplicación sobre los alumnos que la usan por primera vez. Es importante conocer qué opinan los usuarios sobre la organización, la estructura, las figuras, las animaciones y el tratamiento que se les da para que sea fácil el seguimiento del ejercicio.
- Aspectos interactivos: Pretende evaluar todos los elementos relacionados con la navegación y la facilidad para reconocer su funcionamiento.
- Aspectos pedagógicos: Es la parte más importante. Evalúa la cantidad y calidad del contenido de la aplicación. El objetivo principal que busca este proyecto es satisfacer el aprendizaje del alumno, por lo que se medirá directamente este grado de satisfacción.

Finalmente, la matriz creada para la evaluación de la aplicación se muestra en el Anexo III. El formulario hace una distinción entre los alumnos que tienen conocimientos previos de la asignatura y los que no, para realizar un estudio correcto.

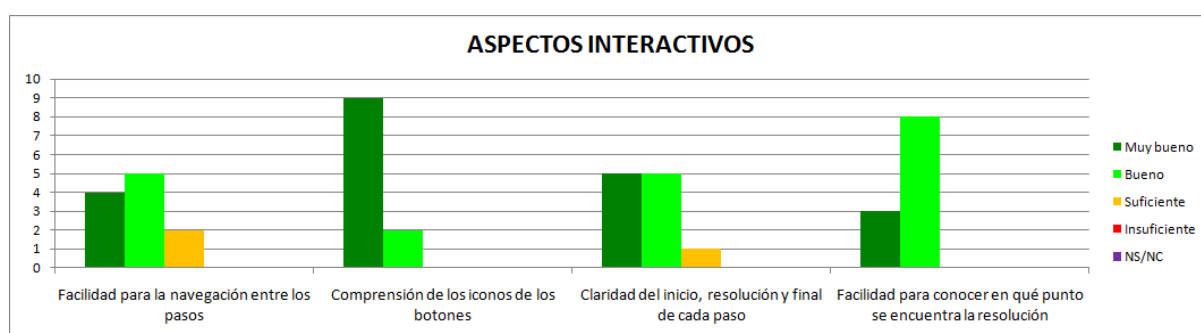
Con la aplicación totalmente desarrollada, se realizó una evaluación de la misma sobre una muestra de 11 usuarios: 4 de ellos sin conocimientos previos y 7 alumnos que ya habían cursado la asignatura. El resultado se puede observar las Figuras 68, 69, 70 y 71.

En la Figura 68 se presenta la evaluación de los aspectos visuales, basada en la opinión de la muestra completa (todos los encuestados). Cabe destacar la gran aceptación de la calidad de las figuras de los problemas y del aspecto general de la aplicación. Por otro lado cuatro de los entrevistados (36,3%) opinan que la capacidad para desviar la atención a los puntos de interés es solo suficiente. Se ha mejorado este aspecto utilizando iluminados más intensos para las animaciones. El resto de los aspectos han sido evaluados positivamente.



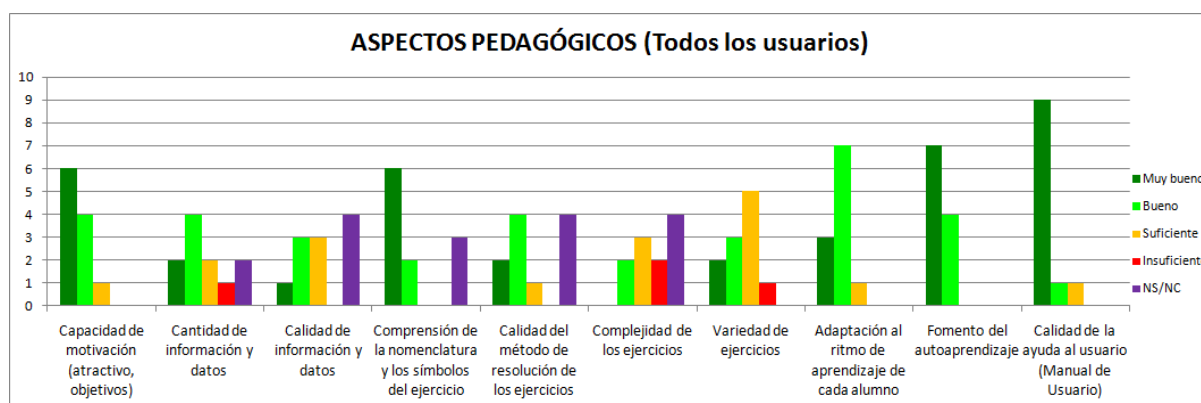
*Figura 68: Evaluación de los aspectos visuales de la aplicación.*

En la Figura 69 se presenta la evaluación de los aspectos interactivos, basada en la opinión de la muestra completa (todos los encuestados). Cabe destacar la fácil comprensión de los iconos de los botones, y la conformidad de los usuarios con la navegación en general.

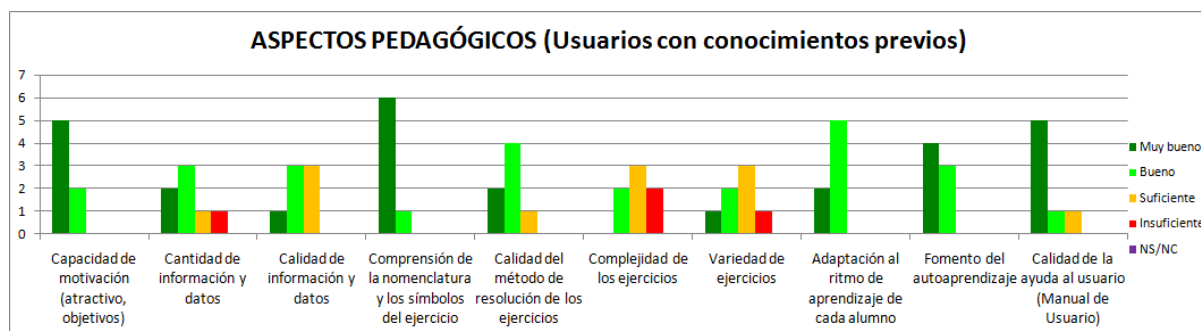


*Figura 69: Evaluación de los aspectos interactivos de la aplicación.*

En la Figura 70 se presenta la evaluación de los aspectos pedagógicos, basada en la opinión de la muestra completa (todos los encuestados). Como puede verse, muchos alumnos no contestan a algunas de las cuestiones, y son precisamente aquellos que no tienen conocimientos previos de la asignatura. Por ello es necesario representar la evaluación basada solo en aquellos usuarios con conocimientos previos (Figura 71).



*Figura 70: Evaluación de los aspectos pedagógicos de la aplicación (Muestra completa, 11 encuestados)*



**Figura 71: Evaluación de los aspectos pedagógicos de la aplicación (Muestra parcial, 7 encuestados con conocimientos previos de la asignatura).**

En el estudio de la Figura 71 es importante mencionar la disconformidad de varios encuestados con los aspectos de: cantidad de información y datos, complejidad de los ejercicios y variedad de ejercicios.

En cuanto a la cantidad de información y datos en los ejercicios, un encuestado (14,3%) opinó que era insuficiente. Puede deberse a que su nivel de conocimiento previo de asignatura fuera más bajo del que se presupone o que considerase que la aplicación ha de funcionar como herramienta de autoaprendizaje y no como un apoyo a las clases de problemas. Para solventarlo, se han completado las explicaciones con los conceptos más importantes de la teoría en la que se fundamentan. Aún así se advierte en la web donde se alojan los ejercicios (<http://maqlab.uc3m.es/TMM/MECMAQ.html>) que la aplicación servirá como apoyo a la docencia.

Observando las opiniones sobre la complejidad de los ejercicios, dos encuestados (28,6%) opinan que es insuficiente. Los ejercicios que se proponen en este proyecto son un ejemplo de cada uno de los bloques de la asignatura que presentan dificultades gráficas en los métodos de resolución. Se ha seleccionado un ejercicio de cada bloque donde se pudieran implementar todos los elementos de animación e interacción. En definitiva, pretenden ser una guía para el desarrollo futuro de más ejercicios.

En relación con el aspecto anterior se puede explicar la disconformidad de un encuestado (14,3%) con la variedad de ejercicios. Este problema se ha solventado estableciendo la metodología para la creación de nuevos ejercicios (Anexo II), con la que se podrá aumentar el repertorio de la aplicación.

Como conclusión final se presenta un diagrama circular por cada uno de los tres aspectos claves para ver el nivel de satisfacción general entre los encuestados (Figuras 72, 73 y 74). Se puede concluir que el nivel de satisfacción es muy alto en los aspectos visuales e interactivos (88% y 93% respectivamente) y que todos los encuestados están al menos conformes con la aplicación. En cuanto a los aspectos pedagógicos se observa que un 4% de los encuestados tiene alguna opinión negativa (un valor muy bajo) y un 15% no contesta debido a que no tiene conocimientos previos de la asignatura. El resto (81%) está al menos conforme con la aplicación.

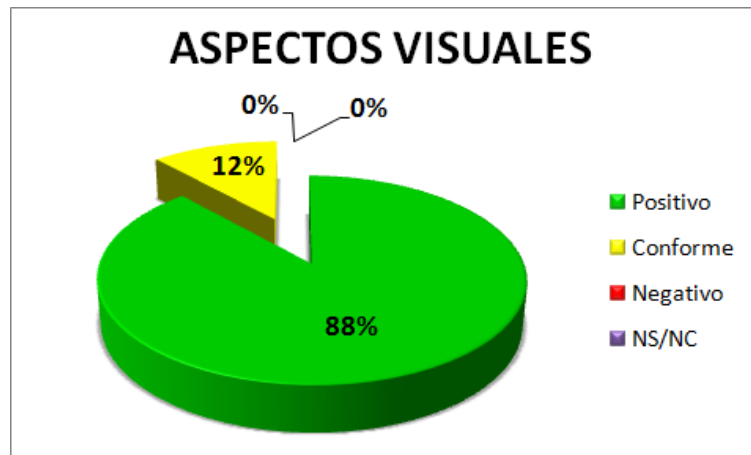


Figura 72: Diagrama circular de satisfacción general con los aspectos visuales de la aplicación

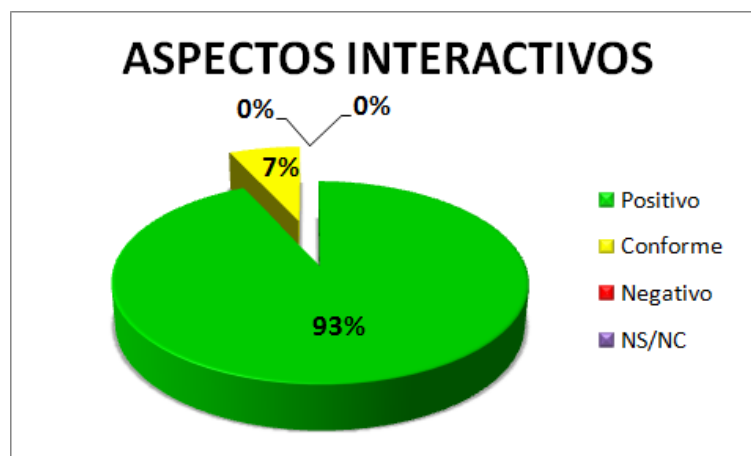


Figura 73: Diagrama circular de satisfacción general con los aspectos interactivos de la aplicación

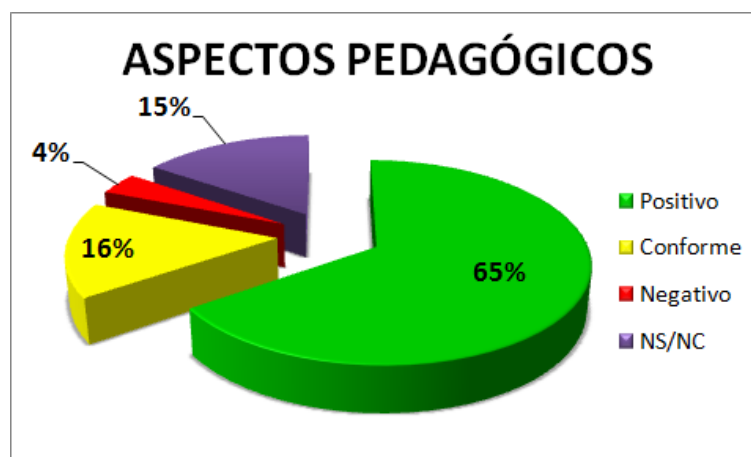


Figura 74: Diagrama circular de satisfacción general con los aspectos pedagógicos de la aplicación

## V) PLANIFICACIÓN

La planificación del proyecto de análisis, diseño y creación de una aplicación para la asignatura de Mecánica de Máquinas se muestra a continuación mediante una tabla de duración de cada fase (Tabla 8) y un diagrama de Gantt (Figura 75).

*Tabla 8: Planificación por fechas del análisis, diseño y creación de la Aplicación de Mecánica de Máquinas*

	Nombre de la tarea	Fecha de inicio	Fecha de finalización	Duración
1	<b>[-] Análisis de Requisitos</b>	07/01/13	08/02/13	25
2	Contenido de la asignatura Mec. de Máquinas	07/01/13	18/01/13	10
3	Análisis de las dificultades	21/01/13	08/02/13	15
4	Estudio del perfil usuario y su entorno	29/01/13	08/02/13	9
5	<b>[-] Diseño de la Aplicación</b>	11/02/13	12/03/13	22
6	Análisis del funcionamiento	11/02/13	08/03/13	20
7	Análisis del entorno y normativa aplicable	18/02/13	12/03/13	17
8	Definición del estilo y presentación	11/02/13	12/03/13	22
9	<b>[-] Implementación</b>	18/03/13	31/05/13	55
10	Creación de la Interfaz de Usuario	18/03/13	10/05/13	40
11	Creación de la navegación	06/05/13	31/05/13	20
12	<b>Prototipado y Evaluación</b>	11/02/13	17/05/13	70
13	<b>[-] Lanzamiento</b>	03/06/13	21/06/13	15
14	Creación de plantillas y producto final	03/06/13	17/06/13	11
15	Creación del presupuesto	17/06/13	21/06/13	5

# Aplicación de Mecánica de Máquinas

Nombre de la tarea	Ene				Feb				Mar					
	Dic 31	Ene 7	Ene 14	Ene 21	Ene 28	Feb 4	Feb 11	Feb 18	Feb 25	Mar 4	Mar 11	Mar 18	Mar 25	Abr 1
<div><div></div> Análisis de Requisitos</div>						Análisis de Requisitos								
Contenido de la asignatura Mec. de Máquinas						Contenido de la asignatura Mec. de Máquinas								
Análisis de las dificultades														
Estudio del perfil usuario y su entorno														
<div><div></div> Diseño de la Aplicación</div>											Diseño de la Aplicación			
Análisis del funcionamiento														
Análisis del entorno y normativa aplicable														
Definición del estilo y presentación														
<div><div></div> Implementación</div>														
Creación de la Interfaz de Usuario														
Creación de la navegación														
Prototipado y Evaluación														
<div><div></div> Lanzamiento</div>														
Creación de plantillas y producto final														
Creación del presupuesto														



## VI) LANZAMIENTO

En este apartado se expondrán los costes económicos derivados de la creación de la aplicación para su lanzamiento al mercado.

El presupuesto de la aplicación para la asignatura de Mecánica de Máquinas, se ha dividido en dos partes: los costes del material utilizado para su creación y los costes de la mano de obra empleada.

Los materiales usados a su vez, se han dividido en material software y material físico (entre los que se encuentra el hardware y la documentación relativa a la normativa).

Material software:

- **Adobe Flash Professional CS5:** Herramienta de autor usada para la creación de la IU y la navegación.
- **Microsoft Word:** Procesador de texto usado para la redacción del presente documento.
- **SmartSheet:** Aplicación web usada para la planificación de los trabajos y su supervisión.

Material hardware y normativa:

- Ordenador con capacidad suficiente para suplir los requerimientos del material software (en especial, los de Adobe Flash Professional CS5)
- Normativa usada para el diseño y creación de la aplicación: ISO/IEC 9126; ISO 9241; ISO/IEC 11581; ISO/IEC 14598; ISO TR 18529.

Adobe Flash Professional CS5 se compra directamente de la página web “<http://www.adobe.com/es/products/flash.html>”. Permite la compra mes a mes, por lo que solo se comprará para los meses de febrero, marzo, abril y mayo, correspondientes con la fase de diseño e implementación de la aplicación. El precio es de 29.99€/mes sin IVA.

Microsoft Word 2013 se compra de forma online desde la página web “<http://www.microsoftstore.com>”. El precio de la licencia es único, de 62.41€ sin IVA para un solo PC.



El software SmartSheet es una aplicación online ([www.smartsheet.com](http://www.smartsheet.com)), que establece un periodo de prueba de 30 días. Para los meses restantes será necesario subscribirse y pagar una cuota de 9.63€/mes para el plan básico.

En cuanto al equipo, se seleccionará conforme a las especificaciones técnicas del software Adobe Flash Professional CS5 que vienen en el propio producto (por ser este el software con los mayores requerimientos)

### Requerimientos de Adobe Flash Professional CS5:

#### Sistema Operativo Windows

- Procesador Intel® Pentium® 4, Intel Centrino®, Intel Xeon®, o Intel Core™ Duo (o compatible)
- Microsoft® Windows® 7 (64 bit) o Windows 8 (64 bit)
- 2GB de RAM (4GB recomendado)
- 2.5GB de espacio en disco duro para la instalación. Se requiere espacio adicional durante la instalación.
- Pantalla con resolución de 1024x768 (1280x800 recomendada)
- Software QuickTime 10.x recomendado
- Conexión a internet necesaria para la activación del producto, la validación del usuario y para el acceso a servicios online.

El equipo seleccionado que cumple estas características es un ordenador Acer Aspire XC600 G645/4GB/500GB/GF 605, con un precio de 318,18€ (sin IVA). La amortización del equipo será de 1.92€ diarios (para un total de 165 días trabajados).

En cuanto a la normativa, los precios se pueden consultar en las páginas de Aenor e ISO y sus precios varían dependiendo de la norma y la parte que interese.

- UNE-ISO/IEC 9126-1:2004 (Ingeniería del software. Calidad del producto software) Precio 39,52€.
- ISO 9241- 151:2008 (Ergonomía de la interacción hombre-sistema; Parte 151: Directrices para las interfaces de usuario web). Precio 53,11€.
- ISO/IEC 11581-1:2011 (Information technology. User interface icons) Precio 53,24€.
- UNE-ISO/IEC 14598-3:2005 (Tecnología de la información. Evaluación del producto software. Parte 3: Procedimiento para desarrolladores.) Precio 35,43€.

- ISO/TR 18529:2000 (Ergonomics of human-system interaction. Human-centered lifecycle process descriptions) Precio 98,41€.

Los costes de mano de obra se contabilizarán dependiendo del rol del profesional usado para la realización del proyecto en cada fase del mismo. Atendiendo al diagrama de Gantt del apartado de planificación (apartado 5, Cap. 4) se determinarán los roles asumidos en cada fase y las horas empleadas por el profesional. Cada empleado trabaja a media jornada, sin embargo, al no tener en cuenta los festivos, se establecerá una media de 3 horas por día de trabajo.

- **Analista**, para la fase de análisis de requisitos. Se emplearon un total de 25 días laborables.
- **Diseñador gráfico**, para la fase de diseño de la aplicación, para la parte de “creación de la interfaz de usuario” dentro de la fase de implementación y para la fase de lanzamiento. Para estas tres partes se emplearon un total de 77 días laborables.
- **Programador**, para la “creación de la navegación” dentro de la fase de implementación. En esta parte se empleó un total de 20 días laborables.
- La fase de prototipado y evaluación entrará dentro de los días del diseñador y el programador ya que se trata de acciones puntuales y no de periodos de tiempos.

El presupuesto desglosado en detalle se muestra en la Tabla 9.

Tabla 9: Presupuesto para la aplicación de Mecánica de Máquinas

Concepto	Total (sin IVA)
<b><u>Material Software</u></b>	<b>219,81 €</b>
Adobe Flash Professional CS5 29,99 €/mes x 4 meses	119,96 €
Microsoft Word 2013 62,41€/licencia	62,41 €
SmartSheet 9,63€/mes x 4 meses	37,44 €
<b><u>Material Hardware</u></b>	<b>318,18 €</b>
Equipo Acer Aspire XC600 57,85 €/mes x 5,5 meses	318,18 €
<b><u>Normativa</u></b>	<b>279,71 €</b>
UNE-ISO/IEC 9126-1:2004 39,52€/licencia	39,52 €
ISO 9241- 151:2008 53,11€/licencia	53,11 €
ISO/IEC 11581-1:2011 53,24€/licencia	53,24 €
UNE-ISO/IEC 14598-3:2005 35,43€/licencia	35,43 €
ISO/TR 18529:2000 98,41€/licencia	98,41 €
<b><u>Mano de Obra</u></b>	<b>5403 €</b>
Analista 20€/hora x 75horas	1500 €
Diseñador Grf. 13€/hora x 231 horas	3003 €
Programador 15€/hora x 60 horas	900 €
<b>Total sin IVA</b>	<b>6220,7 €</b>
<b>IVA (21%)</b>	<b>1306,34 €</b>
<b>TOTAL</b>	<b>7527,04 €</b>

*El presupuesto actual asciende a la cantidad de SIETEMIL QUINIENTOS VEINTISIETE EUROS CON CUATRO CÉNTIMOS (7527,04 €)*

Madrid a 21 de Junio de 2013

# ***CAPÍTULO 5***

## **CONCLUSIÓN Y FUTUROS DESARROLLOS**

---

## I) FUTUROS DESARROLLOS

En este apartado se analizarán las posibles líneas de desarrollo aplicables al producto final de este proyecto. Para ello es necesario realizar un estudio de cómo evolucionarán las tecnologías y la enseñanza y establecer así el desarrollo en la dirección correcta.

Se recurrirá al Informe “*Horizon*” [2] [3] sobre enseñanza universitaria emitido por el Ministerio de Educación, Cultura y Deporte y por el Instituto Nacional de Tecnologías Educativas y de Formación del Profesorado (INTEF). Se atenderá especialmente a las publicaciones de los años 2012 y 2013. Estos informes tienen como objetivo:

*“Identificar las nuevas tecnologías y analizar la repercusión que tendrán en el campo de la enseñanza, el aprendizaje, la investigación y la expresión creativa en los próximos cinco años” [2].*

De manera esquemática, se presentan las tecnologías cuya repercusión se estima en mayor grado en el campo de la educación durante los próximos años.

Tiempo de adopción	Tecnologías	Tiempo de adopción	Tecnologías
Un año o menos	<a href="#">Aplicaciones para dispositivos móviles</a>	Un año o menos	Cursos abiertos masivos en línea o MOOC
	<a href="#">Tabletas</a>		Tabletas
De dos a tres años	<a href="#">Aprendizaje basado en juegos</a>	De dos a tres años	Juegos y Gamificación
	<a href="#">Analíticas de aprendizaje</a>		Analíticas de Aprendizaje
De cuatro a cinco años	<a href="#">Informática basada en gestos</a>	De cuatro a cinco años	Impresión 3D
	<a href="#">Internet de las Cosas</a>		Tecnología portátil

**Figura 76: Previsiones sobre las tecnologías en auge para los próximos cinco años. Informe Horizon 2012 (Izquierda) [2]; Informe Horizon 2013 (derecha) [3].**

Como se puede ver en el informe de Marzo de 2012, para el presente año se produciría un gran desarrollo de las aplicaciones para dispositivos móviles y tabletas. Efectivamente, durante este año, han aumentado exponencialmente el número de aplicaciones para estos dispositivos, entre ellas las aplicaciones dedicadas a la educación y la enseñanza.

Teniendo en cuenta las previsiones realizadas en el informe del 2013, se prestará atención a las líneas de desarrollo hacia los cursos abiertos (MOOC) y las aplicaciones para tabletas y dispositivos portátiles.

Como se definió en el apartado de enseñanza virtual “*e-learning*” (apartado 2.3, Cap. 2), los MOOC han gozado de un reconocimiento público sin precedentes en este último año. Apuestan por una educación continua, más autónoma, y a la medida de las personas. En este sentido, la vía de desarrollo que se propone consiste en:

- Aumentar el número de ejercicios resueltos siguiendo la metodología propuesta en el Anexo II.
- Agregar el contenido teórico de la asignatura completando el ya existente en el dominio web donde se sitúa la aplicación.
- Crear un menú en Flash donde aparezca todo el contenido tanto de ejercicios como de teoría, que funcione como un enlace a ellos.
- Crear un entorno web de la asignatura donde poder hacer preguntas al profesor, proponer nuevos problemas, realizar discusiones de temas relacionados, etc. En definitiva crear un espacio de enseñanza colaborativa de la asignatura.

Por otro lado es importante atender al gran comercio existente en torno a los dispositivos móviles. La aplicación a estudio en este proyecto, es reproducible por los navegadores de los dispositivos que soporten Flash. Hoy en día casi todos traen preinstalados este “*pulg-in*” o se puede instalar, ya que existe mucho contenido web que lo requiere. Aun así se propone:

- La creación de una aplicación móvil con todo el contenido de la web y con la posibilidad de descarga y actualización de nuevos contenidos.
- La aplicación contendría los ejercicios y la teoría de la asignatura, de este modo los alumnos podrían utilizarla durante las clases como apoyo a las explicaciones del profesor.

Estos son algunos de los posibles desarrollos que podrían aplicarse a este proyecto con el fin de dotarle de un mayor alcance a nivel educativo.

## II) CONCLUSIONES

Los cambios en el sistema educativo universitario, y la adaptación de las nuevas tecnologías a la enseñanza, han impulsado la creación de numerosos sistemas interactivos que ayuden al alumno y al profesorado en las labores de aprendizaje y enseñanza.

Ante el problema que presentaba la asignatura de Mecánica de Máquinas a la hora de entender el proceso de resolución de sus ejercicios, se recurrió a las ventajas que ofrecen las tecnologías existentes. Surgió la idea de crear una aplicación que ayudase al alumno a solventar este problema.

La primera observación que habría que señalar es el trabajo personal que he tenido que realizar para encontrar toda la información referente a la creación de aplicaciones interactivas y sus interfaces de usuario. He conseguido adaptar un tema que no se ha visto como tal en la carrera, para conseguir la solución a un problema que yo mismo tuve cuando cursé la asignatura de Mecánica de Máquinas.

De esta forma, la solución fue la creación de un ODEA (Objeto Digital de Enseñanza y Aprendizaje) alojada en un servidor de la universidad, para poder tener acceso a ella vía internet. Se han cumplido todos los objetivos secundarios propuestos al comenzar el proyecto: una interfaz que atraiga al alumno, una navegación sencilla e intuitiva y sobre todo, una explicación dinámica y visual de los ejercicios. Este último punto ha sido el más complicado. He tenido que estudiar la asignatura y entender la resolución de los ejercicios para así conseguir el método de explicación más sencillo posible. Probablemente haya más métodos, sin embargo, el adoptado en la aplicación es sencillo, fácil de visualizar y fácil de entender.

Es importante mencionar también la cantidad de pautas y normativas que existen para diseñar una interfaz de usuario. Antes de empezar con el proyecto, siempre había sido usuario de las aplicaciones. Con la creación de la aplicación, adquirí el punto de vista de diseñador y hasta entonces no había imaginado la labor que hay detrás de una interfaz y la cantidad de normativa y directrices referentes a este tema. El resultado de todo el trabajo, ha sido evaluado muy positivamente por diferentes alumnos de grado en Ingeniería Mecánica. Durante el diseño de la aplicación perdí la visión global del proyecto, por lo que fue necesario elaborar un formulario y realizar una encuesta que me proporcionase los puntos fuertes y débiles de la aplicación.

Como conclusión final del proyecto destacaría la satisfacción personal de haber puesto solución a un problema que yo tuve al cursar la asignatura, y poder ayudar así a otros alumnos que se vean en la situación en la que yo me vi.

# BIBLIOGRAFÍA

- [1] Irene Suárez Bravo, "Aplicación de nuevas tecnologías orientadas al aprendizaje de Teoría de Máquinas y Mecanismos," Leganés, 2005.
- [2] Instituto Nacional de Tecnologías Educativas y de Formación del Profesorado (INTEF), "Informe Horizon 2012: Enseñanza Universitaria," Departamento de Proyectos Europeos, Ministerio de Educación Cultura y Deporte, España, Marzo 2012.
- [3] Instituto Nacional de Tecnologías Educativas y de Formación del Profesorado (INTEF), "Horizon Report: 2013 Higher Education Edition," Departamento de Proyectos Europeos, Ministerio de Educación Cultura y Deporte, España, Mayo 2013.
- [4] Gloria Josefina Rosique Cedillo and Jose Antonio Rosique, "El proceso de Bolonia: los desafíos de la convergencia europea en instituciones de educación superior en España," *Redalyc*, vol. 14, no. Nº 30, pp. 231-245, enero-abril 2011, Disponible en: <http://www.redalyc.org/src/inicio/ArtPdfRed.jsp?iCve=67618934015>.
- [5] "Real decreto por el que se establece la estructura de las enseñanzas universitarias y se regulan los estudios universitarios oficiales de grado," Gobierno de España, 29.05.03.
- [6] (2012, Octubre) Wikipedia. [Online]. [www.wikipedia.es](http://www.wikipedia.es) fecha última visita (24-2-2013)
- [7] Mario de Miguel Diaz, "Adaptación de la homologación de los planes de estudio a la convergencia europea," Dirección general de Universidades, Ministerio de Educación, Cultura y Deporte, Madrid, Octubre, 2004.
- [8] Guillermina Waldegg Casanova, "El uso de las nuevas tecnologías para la enseñanza y el aprendizaje de las ciencias," *Revista electrónica de Investigación Educativa REDALYC*, vol. 4, no. 1, pp. 95-116, Mayo 2002.
- [9] J Uceda and S Barro, "Las TIC en el Sistema Universitario Español," Madrid, 2008.
- [10] Dr. Santiago Almeida Campos, "Enseñanza Asistida por Ordenador," in *Facultad de Ciencias Médicas*, Matanzas, Cuba, 26 de Marzo de 1997.
- [11] Departamento de Lenguajes y Sistemas Informáticos. (Curso 2006-2007) Nuevas tecnologías aplicadas a educación. Archivo ppt.
- [12] Microsoft. Microsoft Silverlight. [Online]. <http://msdn.microsoft.com/es-es/silverlight/bb187401.aspx> fecha última visita (8-11-2012)
- [13] Apple. Tumult Hye 1.6; Create, Animate, HTML5. [Online]. <http://tumult.com/hype/> fecha última visita (9-11-2012)
- [14] M<sup>a</sup> Dolores Parra Sageras and Carmen Luengo San José, "Squeak, la programación del siglo XXI," *UNED, XI Congreso Internacional Informática Educativa*.
- [15] Juan de Dios, "Tutorial de Macromedia Flash MX," Madrid, 2004.
- [16] Instituto de Tecnologías Educativas. Utilización didáctica de la Pizarra digital. [Online]. [http://www.ite.educacion.es/formacion/materiales/137/cd/m4\\_recursos\\_educativos/programas\\_de\\_autor.html](http://www.ite.educacion.es/formacion/materiales/137/cd/m4_recursos_educativos/programas_de_autor.html) fecha última visita (3-12-2012)
- [17] Stat Owl. (2012, Sep.) es.wikipedia.org. [Online]. [http://www.statowl.com/custom\\_ria\\_market\\_penetration.php](http://www.statowl.com/custom_ria_market_penetration.php) fecha última visita (10-12-2012)
- [18] Sham Bhangal and Ben Renow-Clarke, *Fundamentos de ActionScript para Flash MX*. Madrid, España: Anaya Multimedia, 2003.
- [19] Carlos Marcelo García, *E-learning teleformación: diseño, desarrollo y evaluación de la formación a través de Internet.*: Editorial Gestión 2000 S.A., 2001.



- [20] Julio Cabero Almenara, "Bases pedagógicas del e-learning," vol. 3, no. Nº1, Abril 2006.
- [21] MOOC.es. (2013) MOOC. [Online]. [www.MOOC.es](http://www.MOOC.es) fecha última visita (2-2-2013)
- [22] Telefónica Learning Services, Universia Holding. (2012) MiriadaX. [Online]. <http://miriadax.net/> fecha última visita (5-2-2013)
- [23] Coursera. [Online]. <https://www.coursera.org/> fecha última visita (5-2-2013)
- [24] Khan Academy, Inc. (2013) Khan Academy. [Online]. <https://www.khanacademy.org/> fecha última visita (5-2-2013)
- [25] Udacity, Inc.. (2013) Udacity. [Online]. <https://www.udacity.com/> fecha última visita (5-2-2013)
- [26] Harvard and MIT. (2013) edX. [Online]. <https://www.edx.org/> fecha última visita (5-2-2013)
- [27] Alejandrino Gallego Rodríguez and Eva Martínez Caro, "Estilos de aprendizaje y e-learning. Hacia un mayor rendimiento académico," Departamento de Economía de la Empresa, Universidad Politécnica de Cartagena,.
- [28] Eva Fernández Gómez, *E-learning: implantación de proyectos de información On-Line.*: Editorial Microinformática, 2003, vol. I.
- [29] Francesc Esteve Mon, "Bologna y las TIC: De la docencia 1.0 al aprendizaje 2.0," *La Cuestión Universitaria*, no. 5, pp. 59-68, 2009.
- [30] María Belén Vaquerizo-García, "Enseñanza-Aprendizaje con Web 2.0 y 3.0," *Revista de Comunicación Vivat Academia*, no. Nº Especial, pp. 116-121, Febrero 2012.
- [31] Susana Martínez Naharro, Pilar Bonet Espinosa, Pilar Cáceres González, Fernando Fargueta Cerdá, and Eloína García Felix, "Los objetos de aprendizaje como recurso de calidad para la docencia: criterios de validación de objetos en la Universidad Politécnica de Valencia," Area de Sistemas de Información y Comunicaciones, Instituto de Ciencias de la Educación. Univ. de Valencia, España,.
- [32] JCA Solutions. The SCORM authority. [Online]. <http://www.scormsoft.com/scorm> fecha última visita (7-3-2013)
- [33] Beatriz Osorio, Jaime Muñoz, Rodrigo Álvarez, and Carlos Arévalo, *Metodología Para Elaborar Objetos de Aprendizaje e Integrarlos a un Sistema de Gestión de Aprendizaje.*, 2005.
- [34] Miguel Gea and Fco. Luis Gutiérrez, "La interacción Persona-Ordenador. El diseño," Universidad de Granada, 2002.
- [35] M. Gómez Leopoldo Sebastián, "Diseño de Interfaces de Usuario: Principios, Prototipos y Heurísticas para Evaluación," Argentina,.
- [36] Benjamín Roe, "Diseño de Interfaces de Usuario Usables: Una Guía Rápida para Desarrolladores de Software Libre y de Código Abierto," 2004.
- [37] Yusef Hassan Montero and Sergio Ortega Santamaría, "Informe APEI sobre Usabilidad," Asociación Profesional de Especialistas en Información, Ministerio de Cultura, Gijón, Informe APEI 3 ISBN: 978-84-692-3782-3, 2009.
- [38] B. Shneiderman, *Designing the user interface. Strategies for effective human computer interaction*, 3rd ed. MA: Addison and Wesley, 1990.
- [39] Wikipedia. [Online]. [www.wikipedia.es](http://www.wikipedia.es) fecha última visita (15-3-2013)
- [40] Jesús Tramullas Saz, "El diseño centrado en el usuario para la creación de productos y servicios de información digital," Departamento de Ciencias de la Documentación, Zaragoza,.
- [41] Smith W. J., "ISO and ANSI ergonomic standards for computer products," 1985.
- [42] Ana Belen Martínez and Juan Manuel Cueva, "La interacción Persona-Ordenador. Estándares y guías," Universidad de Oviedo, 2001.
- [43] Cristina Castejón Sisamón, "Teoría de Máquinas," Universidad Carlos III, Madrid, Proyecto docente e investigador 2009.
- [44] Juan Carlos García Prada, Cristina Castejón Sisamón, and Higinio Rubio Alonso, *Problemas Resueltos de Teoría de Máquinas y Mecanismos*. Madrid, España: Thomson Editores Spain, 2007.

- [45] Irene Suárez Bravo, "Aplicación de Nuevas Tecnologías Orientadas al Aprendizaje de Teoría de Máquinas y Mecanismos," Departamento de Ingeniería Mecánica, Universidad Carlos III, Madrid, Proyecto Fin de Carrera 2005.
- [46] Microsoft Corporation, "The Microsoft Windows User Experience. Official Guidelines for User Interface Developers and Designers," Guía de estilo.
- [47] Sebastián Torres Toro, *Metodología de la enseñanza abierta y a distancia: (e-learning)*.: Editorial La Montaña, 2001.
- [48] María Jesús Fernández Ruiz, José María Angós Ullate, and José Antonio Salvador Oliván, "Interfaces de Usuario: Diseño de la visualización de la información como medio para mejorar la gestión del conocimiento y los resultados obtenidos por el usuario," V Congreso ISKO, España,.
- [49] Fernando Olivera Domingo, "Interfaces de Usuario Laterales: Estudio y Desarrollo de Tecnologías para Permitir la Interacción Sutil," Universidad Autónoma de Madrid, Madrid, Trabajo de Fin de Master Junio 2011.

### Normativa y estándares

ISO/IEC 9126	"Evaluación de productos software – características de calidad y directrices para su uso".
ISO 9241	"Requisitos ergonómicos para trabajar con terminales de presentación visual ( <i>Visual Display Terminals, VDTs</i> )". Parte 11.
ISO 11064	"Diseño ergonómico de centros de control". Parte 7: Principios para la evaluación de los centros de control.
ISO/IEC 14598	"Evaluación de productos software".
ISO 13407	"Procesos de diseño centrados en la persona para sistemas interactivos".
ISO/IEC 10741	"Interacción de diálogos".
ISO/IEC 11581	"Símbolos y funciones de los iconos".
ISO 13406	"Requisitos ergonómicos para trabajar con presentaciones visuales basadas en paneles planos".
ISO/IEC 14754	"Gestos comunes para la edición de textos con sistemas basados en lápiz".
ISO/IEC 15910	"Proceso de documentación de software de usuario".
ISO TR 18529	"Descripciones de los procesos del ciclo de vida centrados en la persona".
ISO 10075	"Principios ergonómicos relacionados a la sobrecarga mental".
ISO 14915	"Ergonomía del software para interfaces de usuario multimedia".

# ***ANEXO I***

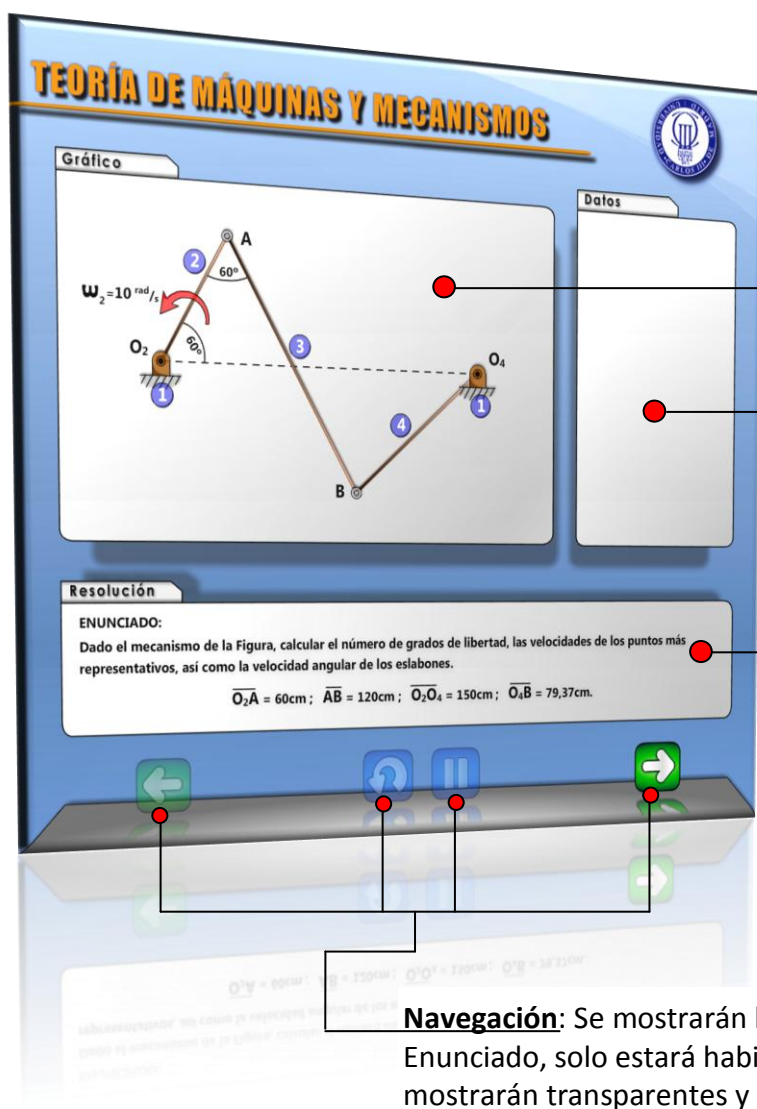
## **MANUAL DE USUARIO**

---

En el presente documento se detalla toda la información necesaria para comprender las diferentes funcionalidades de la aplicación de Mecánica de Máquinas. Para ello se estructurará el manual en tres secciones que se corresponden con las tres fases posibles en la resolución de un problema: Enunciado, resolución del paso, final del paso y final del ejercicio.

## ENUNCIADO

Se presenta la estructura de la interfaz y los diferentes elementos que se pueden encontrar en el enunciado:



**Panel Gráfico:** Se muestra la estructura de la que parte el problema. Aparecerán los datos como vectores y ángulos que requieran representación gráfica.

**Panel Datos:** Se mostrarán los resultados obtenidos a medida que se resuelve el problema.

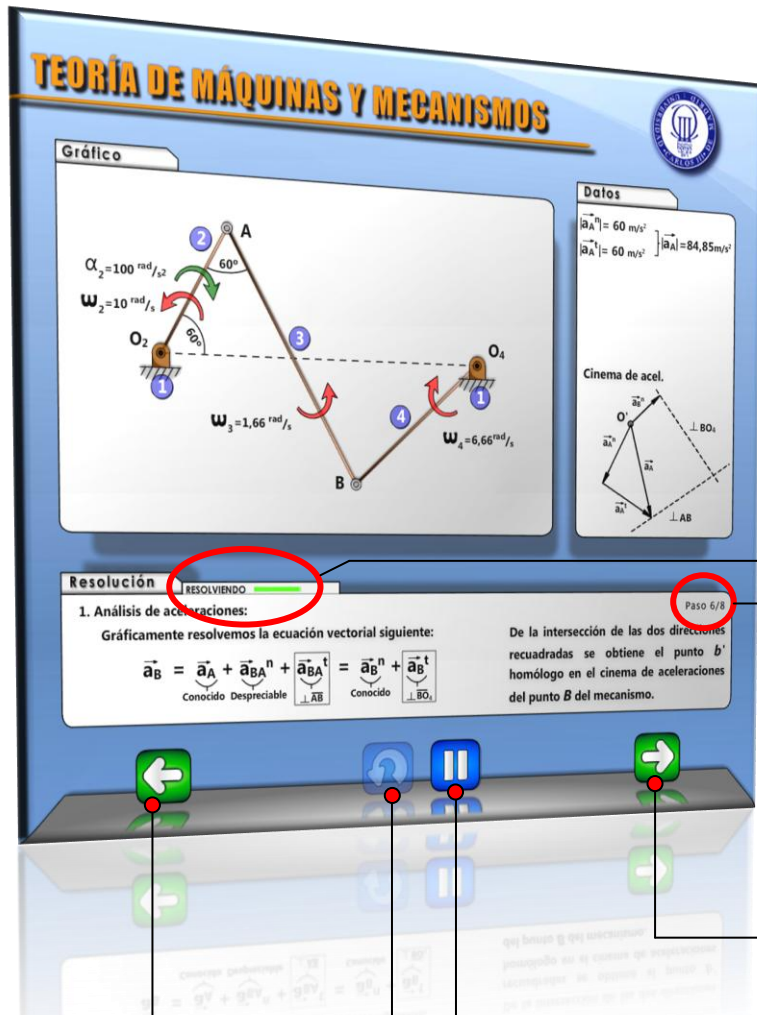
**Panel Resolución:** Al comienzo se mostrará el enunciado y los datos necesarios para resolver el problema.

**Navegación:** Se mostrarán los botones de navegación. En el Enunciado, solo estará habilitado el botón de "Siguiente". El resto se mostrarán transparentes y no será posible hacer *click* sobre ellos.

**ENUNCIADO:**  
 Dado el mecanismo de la Figura, calcular el número de grados de libertad, las velocidades de los puntos más representativos, así como la velocidad angular de los eslabones.  
 $\overline{O_2A} = 60\text{cm}$ ;  $\overline{AB} = 120\text{cm}$ ;  $\overline{O_2O_4} = 150\text{cm}$ ;  $\overline{O_4B} = 79,37\text{cm}$ .

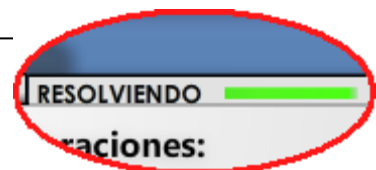
# RESOLUCIÓN

En este apartado se detallan todos los elementos con los que el usuario puede interactuar durante la resolución de un paso

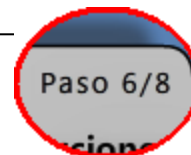


The screenshot shows the 'TEORÍA DE MÁQUINAS Y MECANISMOS' software interface. It includes a 'Gráfico' section with a mechanism diagram, a 'Datos' section with numerical values, and a 'Resolución' section with a list of steps. The 'Resolución' section is currently on 'Paso 6/8'. The interface also features a 'Barra de Estado' at the top and a 'Siguiendo' button at the bottom right.

**Barra de Estado:** Significa que el paso se está resolviendo. Desaparecerá cuando haya terminado la explicación del paso



**Indicador de Paso:** Muestra el paso actual frente al número total de pasos.



**Siguiente (\*):** Avanza directamente hasta el paso siguiente.

**Anterior (\*):** Retrocede al paso previo (o al enunciado si está en el paso 1).

**Repetir:** (Bloqueado) Hay que esperar a finalizar el paso para poder repetir la explicación.

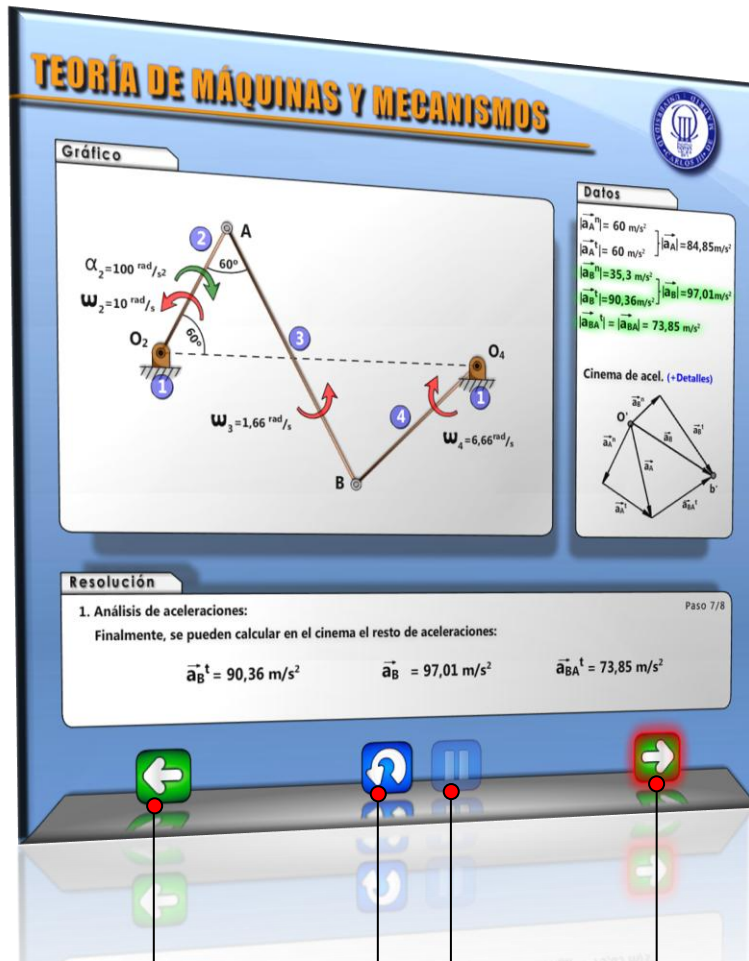
**Pause:** Detiene la reproducción del paso. La barra de Estado se detiene como indicativo de pausa. Para continuar habrá que pulsar el botón "Reanudar" que aparecerá en el mismo lugar.



(\*): No hará falta esperar a que termine la resolución del paso para avanzar o retroceder.

# FINAL DEL PASO

En este apartado se mostrarán los elementos que aparecen cuando finaliza la resolución de un paso.



**Datos:** En la sección datos, aparecerán iluminados en verde aquellos datos que se hayan obtenido en el paso actual.

$$|\vec{a}_{BA}^t| = |\vec{a}_{BA}| = 73,85 \text{ m/s}^2$$

**Indicador de avance:** El botón "Siguiente" se iluminará para indicar que ha finalizado la resolución del paso y se puede continuar.



**Siguiente:** Avanza directamente hasta el paso siguiente.

**Anterior:** Retrocede al paso previo (o al enunciado si está en el paso 1).

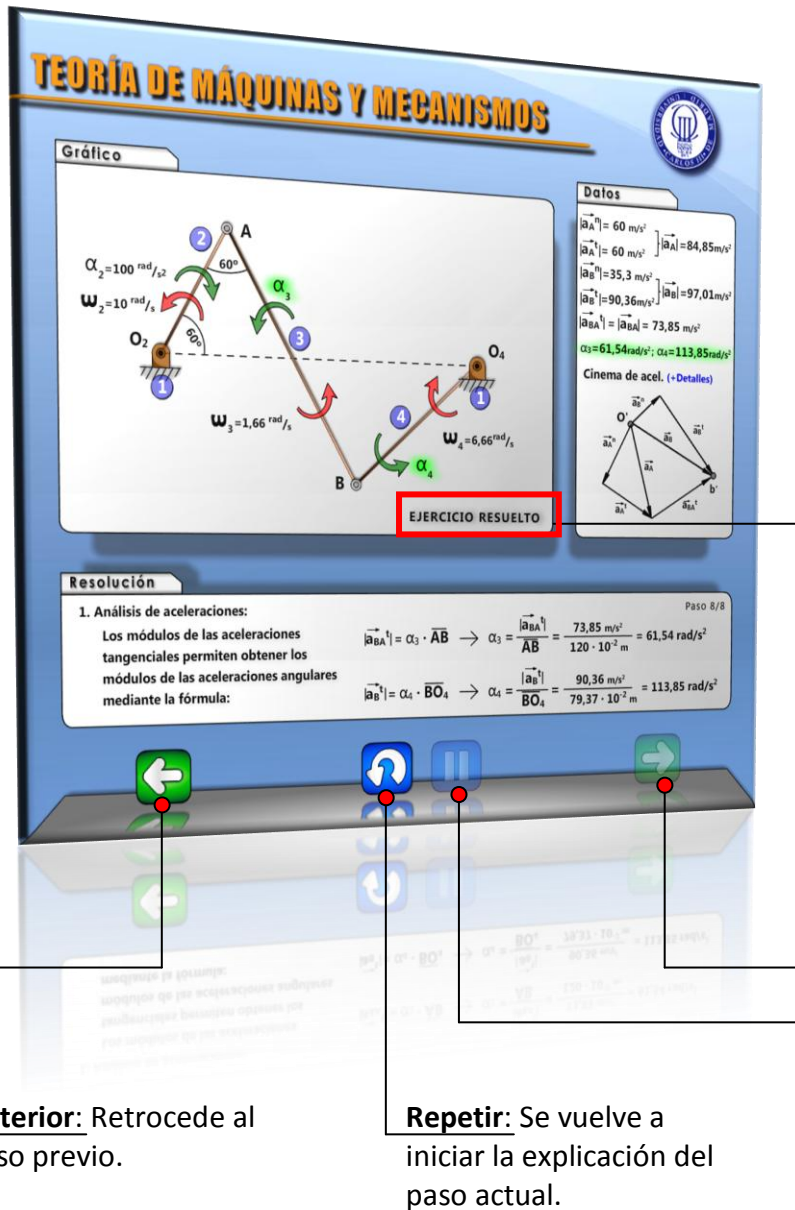
**Repetir:** Se vuelve a iniciar la explicación del paso actual.

**Pause:** (Bloqueado) solo estará habilitado durante la explicación de un paso



# FINAL DEL EJERCICIO

En este apartado se muestran los elementos que aparecen al finalizar el ejercicio por completo.



**TEORÍA DE MÁQUINAS Y MECANISMOS**

**Gráfico**

Diagrama de un mecanismo con los siguientes datos:

- $\alpha_2 = 100 \text{ rad/s}^2$
- $\omega_2 = 10 \text{ rad/s}$
- $\omega_3 = 1,66 \text{ rad/s}$
- $\omega_4 = 6,66 \text{ rad/s}$

**Datos**

$ \vec{a}_A  = 60 \text{ m/s}^2$	$ \vec{a}_B  = 60 \text{ m/s}^2$	$ \vec{a}_A  = 84,85 \text{ m/s}^2$
$ \vec{a}_B  = 35,3 \text{ m/s}^2$	$ \vec{a}_B  = 90,36 \text{ m/s}^2$	$ \vec{a}_B  = 97,01 \text{ m/s}^2$
$ \vec{a}_{BA}  = 73,85 \text{ m/s}^2$	$\alpha_3 = 61,54 \text{ rad/s}^2$	$\alpha_4 = 113,85 \text{ rad/s}^2$

**Cinema de acel. (+Detalles)**

**Resolución**

1. Análisis de aceleraciones:

Los módulos de las aceleraciones tangenciales permiten obtener los módulos de las aceleraciones angulares mediante la fórmula:

$$|\vec{a}_{BA}| = \alpha_3 \cdot \overline{AB} \rightarrow \alpha_3 = \frac{|\vec{a}_{BA}|}{\overline{AB}} = \frac{73,85 \text{ m/s}^2}{120 \cdot 10^{-2} \text{ m}} = 61,54 \text{ rad/s}^2$$

$$|\vec{a}_B| = \alpha_4 \cdot \overline{BO_4} \rightarrow \alpha_4 = \frac{|\vec{a}_B|}{\overline{BO_4}} = \frac{90,36 \text{ m/s}^2}{79,37 \cdot 10^{-2} \text{ m}} = 113,85 \text{ rad/s}^2$$

**EJERCICIO RESUELTO**

**Anterior:** Retrocede al paso previo.

**Repetir:** Se vuelve a iniciar la explicación del paso actual.

**Pause:** (Bloqueado) solo estará habilitado durante la explicación de un paso

**Siguiente:** (Bloqueado) No hay más pasos.

# ***ANEXO II***

## **METODOLOGÍA DE CREACIÓN DE EJERCICIOS**

---



En el presente documento se detalla la metodología de creación de los ejercicios de mecánica de máquinas con la herramienta de autor Adobe Flash Professional para aumentar el repertorio de la aplicación interactiva expuesta en el proyecto. Se presentan la plantilla y los diferentes elementos que la forman, así como el código necesario para implementar la navegación. Este documento se presenta como una guía de estilo de la aplicación.

Se dividirá en dos partes: la primera referente al diseño gráfico de la interfaz y la segunda a su navegación.

## INTERFAZ DE USUARIO

Para la creación de la interfaz de usuario, se proporciona una plantilla en formato “.fla” (archivo de *flash*) que contiene los elementos comunes a todos los ejercicios y el código de la navegación. Esta plantilla está organizada en carpetas que contienen las capas con los diferentes elementos de la interfaz (Figura 77).

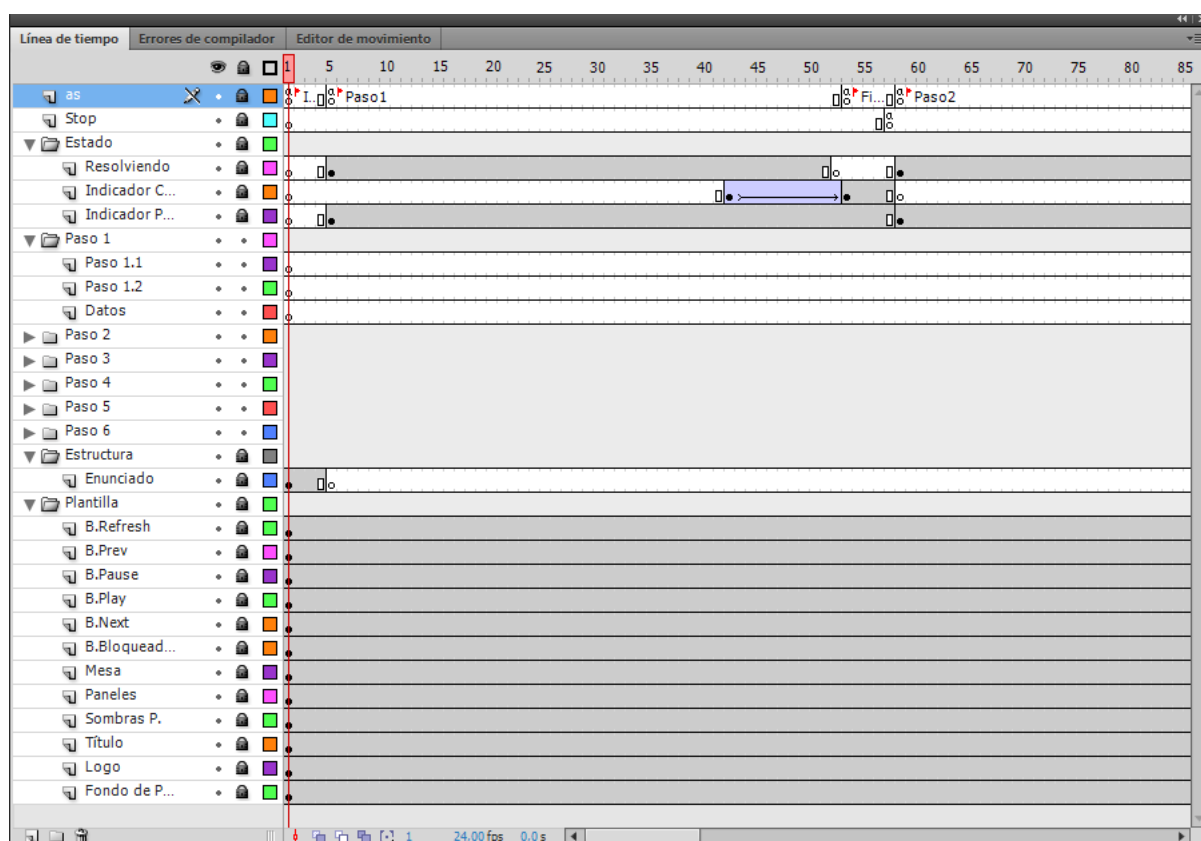


Figura 77: Organización de las capas y la línea del tiempo de la plantilla para los problemas.

La capa “as” y “stop” contienen el código de la navegación que será explicado en el siguiente apartado.

La carpeta “Estado” contiene los elementos que informan al usuario del progreso y la situación de la explicación. No será necesario modificarlos.

A continuación se presentan los pasos de la explicación separados por carpetas. En ellos, el creador del ejercicio ha de insertar las capas necesarias y concluir con los datos obtenidos. Cada paso comenzará en un fotograma concreto y terminará en otro que a su vez coincidirán con los fotogramas “PasoX” y “Fin\_PasoX” de la capa “as” (donde X será el número del paso).

En la carpeta “Estructura” el creador deberá construir la estructura del enunciado separándola en diferentes capas. Para ello se ponen a su disposición una serie de archivos Flash que contienen los elementos comunes a todas las estructuras, con el fin de mantener el mismo estilo en todos los ejercicios de la aplicación. Los archivos son los siguientes:

- *Apoyo fla*: Contiene el elemento de apoyo de la estructura.
- *Barra fla*: Contiene una barra con una longitud estándar. Deberá ser acortada, alargada o girada para adecuarla a la nueva estructura.
- *Unión fla*: Contiene la rótula de unión entre barras.
- *Num. Elemento fla*: Contiene el símbolo para numerar cada elemento de la estructura.
- *V angular fla*: Contiene el vector de velocidad angular y aceleración angular para ejes que salen del plano.
- *CIR fla*: contiene la nomenclatura usada para identificar los Centros Instantáneos de Rotación.

Por último se encuentra la carpeta “Plantilla” donde se distribuirán por capas los elementos de la interfaz. No será necesario modificar esta carpeta. Los elementos que contiene son:

- El título de la asignatura.
- El fondo de la interfaz.
- El logotipo de la Universidad Carlos III.
- Los paneles con sus respectivas sombras.
- La “mesa” o región donde se emplazarán los botones.
- Los botones en formato bloqueado (con transparencia).

- Todos los botones para la navegación.

Estos elementos se distribuyen en capas diferentes y ordenadas según su posición en la interfaz (los elementos de arriba ocuparán las primeras posiciones).

El creador tendrá que desarrollar el ejercicio en las carpetas de: “Estructura”, “Paso 1”, “Paso 2”, “Paso 3”, etc. Podrá incluir todos los pasos que necesite o eliminar los sobrantes.

La cantidad de fotogramas por paso es variable. Si fuese necesario, se añadirán o eliminarán fotogramas.

En la capa de “enunciado”, dentro de la carpeta “Estructura”, se añade un texto que servirá de patrón para todas las explicaciones del ejercicio. El tamaño de letra variará de 12 puntos para texto normal a 14 puntos para letras, símbolos y nomenclatura del ejercicio. El resto de las propiedades se mantendrán igual.

Las animaciones de aparición de textos y elementos serán de entre 5 y 10 fotogramas de duración con una interpolación clásica. En el primer fotograma se aplicará un desenfoque con las siguientes características:

- Desenfoque X: 10 px.
- Desenfoque Y: 50 px.
- Calidad: alta.

Los datos al final de un paso se resaltaran con un iluminado que llame la atención del usuario. Al ser un texto, se animará como anteriormente se ha explicado, colocando en el último fotograma de la animación un iluminado con las siguientes características:

- Desenfoque X: 20 px.
- Desenfoque Y: 10 px.
- Intensidad: 200% (podrá aumentarse si se trata de iluminar un vector)
- Calidad: alta.
- Color: #00FF00 (verde)

Para resaltar un elemento o un texto fuera del campo de percepción visual del usuario y desviar su atención, se utilizará un iluminado más intenso con una animación más rápida. Por lo general se utilizará una animación en 10 fotogramas con el iluminado en el fotograma

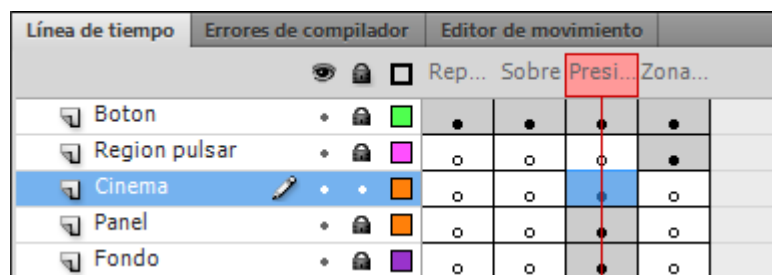
de en medio y con una interpolación clásica. El iluminado ha de tener las siguientes características:

- Desenfoque X: 20 px.
- Desenfoque Y: 20 px.
- Intensidad: 500% (podrá aumentarse dependiendo del tamaño del objeto iluminado)
- Calidad: alta.
- Color: #FF0000 (rojo)

Los movimientos de los vectores u otros elementos serán fluidos con una parte rápida al inicio de la animación y más lenta al final, evitando que detengan el movimiento bruscamente.

Se podrá aplicar transparencia a las partes de la estructura que molesten para la explicación del paso, sin que lleguen a desaparecer. Para ello se les dará un efecto de color, alfa del 10%.

En el caso de que el cinema no quepa en el panel de datos o su información no se interprete bien, se proporciona al creador un botón (en el archivo “+detalles fla”) que al ser pulsado, abre un panel más grande por encima del resto. Se trata de un botón en el que el creador solo ha de modificar el fotograma marcado en azul en la siguiente figura (Figura 78). En dicho fotograma se colocará el cinema detallado, el resto no se modificarán..



*Figura 78: Línea del tiempo del botón “+detalles”.*

El resto de animaciones y elementos que incorpore el creador en los nuevos ejercicios, han de seguir el estilo de los ya creados. El propósito es que el alumno se sienta en la misma aplicación con cada ejercicio y que pueda interiorizar el funcionamiento a lo largo de su experiencia.

# NAVEGACIÓN

En este apartado se detalla el procedimiento para implementar la navegación en la creación de nuevos ejercicios. En la plantilla se ha incorporado todo el código en las dos primeras capas: capa “as” y “stop”.

La capa “as” (*action-script*) está dividida en intervalos de fotogramas que establecen el comienzo y el final de cada paso. Los fotogramas con una “a” inscrita, poseen código.

La capa “stop” sirve únicamente para detener la animación al final de cada paso.

El usuario solo tendrá que modificar el código cuando el número de pasos requeridos para el ejercicio, no concuerde con el número de pasos de la plantilla (6 pasos).

En este caso, los cambios a realizar serán los siguientes:

- **Fotograma 1 de la capa “as”:** a partir de la línea 56, se muestran las funciones para ir a los diferentes pasos. El creador deberá añadir o eliminar una función por cada paso.

```
function IrPaso1(e:MouseEvent):void
{
    gotoAndPlay("Paso1");
}
```

Habrán tantas funciones de este tipo como número de pasos tenga el ejercicio.

- **Primer fotograma del último paso:** se copiará directamente el siguiente código, teniendo en cuenta que:

- X: último paso.
- [X-1]: penúltimo paso.
- [X-2]: antepenúltimo paso.

```
//Durante la Reproducción del Paso
import flash.events.MouseEvent;

//Botones Activos e Inactivos
siguiente_btn.enabled = false;
anterior_btn.enabled = true;
refresh_btn.enabled = false;

//Botones visibles
siguiente_btn.visible = false;
anterior_btn.visible = true;
```

```
refresh_btn.visible = false;
pause_btn.visible = true;
play_btn.visible = false;

//Quitamos los listeners del Paso [X-1]
siguiente_btn.removeEventListener (MouseEvent.CLICK,
IrPasoX);
anterior_btn.removeEventListener (MouseEvent.CLICK,
IrPaso[X-2]);
refresh_btn.removeEventListener (MouseEvent.CLICK,
IrPaso[X-1]);

//Acciones durante la Reproducción del paso
anterior_btn.addEventListener (MouseEvent.CLICK, IrPaso[X-
1]);
```

- **Último fotograma del ejercicio:** se copiará directamente el siguiente código, teniendo en cuenta que X es el último paso:

```
//Botones Activos e Inactivos
siguiente_btn.enabled = false;
anterior_btn.enabled = true;
refresh_btn.enabled = true;

//Botones visibles
siguiente_btn.visible = false;
anterior_btn.visible = true;
refresh_btn.visible = true;
pause_btn.visible = false;
play_btn.visible = false;

//Acciones de botones
refresh_btn.addEventListener (MouseEvent.CLICK, IrPasoX);
```

- Para programar el **resto de pasos** habrá que seguir el siguiente procedimiento. Se copiará el siguiente código teniendo en cuenta que:

- X: paso actual.
- [X-1]: paso anterior.
- [X-2]: dos pasos anteriores.
- [X+1]: paso siguiente.

Código de inicio del paso:

```
//Durante la Reproduccion del Paso
```

```
import flash.events.MouseEvent;

//Botones Activos e Inactivos
siguiente_btn.enabled = true;
anterior_btn.enabled = true;
refresh_btn.enabled = false;

//Botones visibles
siguiente_btn.visible = true;
anterior_btn.visible = true;
refresh_btn.visible = false;
pause_btn.visible = true;
play_btn.visible = false;

//Quitamos los listeners del Paso[X-1]
siguiente_btn.removeEventListener (MouseEvent.CLICK,
IrPasoX);
anterior_btn.removeEventListener (MouseEvent.CLICK,
IrPaso[X-2]);
refresh_btn.removeEventListener (MouseEvent.CLICK,
IrPaso[X-1]);

//Acciones durante la Reproducción del paso
siguiente_btn.addEventListener (MouseEvent.CLICK,
IrPaso[X+1]);
anterior_btn.addEventListener (MouseEvent.CLICK,
IrPaso[X-1]);
```

Código de fin del paso

```
//Botones Activos e Inactivos
siguiente_btn.enabled = true;
anterior_btn.enabled = true;
refresh_btn.enabled = true;

//Botones visibles
siguiente_btn.visible = true;
anterior_btn.visible = true;
refresh_btn.visible = true;
pause_btn.visible = false;
play_btn.visible = false;

//Acciones de botones
refresh_btn.addEventListener (MouseEvent.CLICK,
IrPasoX);
```

# ***ANEXO III***

## **FORMULARIO DE EVALUACIÓN**

---



## FORMULARIO DE EVALUACIÓN DE SOFTWARE EDUCATIVO

### *Información General*

**Nombre o Título del Programa:** Aplicación Interactiva para la resolución de problemas de Mecánica de Máquinas.

**Tipo de programa:** Objeto Digital de Enseñanza y Aprendizaje (ODEA) de ejercicios de repetición.

**Autor(es):** Javier Miranda Puente.

**Temática:** Mecánica de Máquinas.

### *Formulario*

¿Tiene conocimientos previos de Mecánica de máquinas?      SI ☐      NO ☐

#### 1) Evaluación de los aspectos visuales

Criterio	Muy Bueno	Bueno	Suficiente	Insuficiente	No sé / No contesto
Aspecto general de la aplicación					
Estructura visual y organización del contenido					
Calidad de las figuras de los problemas					
Legibilidad y comprensión del texto					
Calidad de las animaciones					
Facilidad en el seguimiento de los problemas					
Capacidad para desviar la atención a los puntos de interés					

## 2) Evaluación de los aspectos interactivos

<b>Criterio</b>	<b>Muy Bueno</b>	<b>Bueno</b>	<b>Suficiente</b>	<b>Insuficiente</b>	<b>No sé / No contesto</b>
Facilidad para la navegación entre los pasos					
Comprensión de los iconos de los botones					
Claridad del inicio, resolución y final de cada paso					
Facilidad para conocer en qué punto se encuentra la resolución					

## 3) Evaluación de los aspectos pedagógicos

<b>Criterio</b>	<b>Muy Bueno</b>	<b>Bueno</b>	<b>Suficiente</b>	<b>Insuficiente</b>	<b>No sé / No contesto</b>
Capacidad de motivación (atractivo, objetivos)					
Cantidad de información y datos					
Calidad de información y datos					
Comprensión de la nomenclatura y los símbolos del ejercicio					
Calidad del método de resolución de los ejercicios					
Complejidad de los ejercicios					
Variedad de ejercicios					
Adaptación al ritmo de aprendizaje de cada alumno					
Fomento del autoaprendizaje					
Calidad de la ayuda al usuario (Manual de Usuario)					